Wavelet analysis on financial time series

By
Arlington Fonseca Lemus

Tutor
Hugo Eduardo Ramirez Jaime

A thesis submitted in partial fulfillment for the degree of
Master in Quantitative Finance

Universidad del
Rosario

Faculty of Economics

July 2018

Abstract

Wavelet methods possess some features that make them a tool with great potential for financial research. The purpose of this thesis is to study the usefulness of wavelet methods in financial time series analysis, for which data from Colombian financial market has been used.

In this thesis the wavelet theory is briefly presented, with a special focus on the Discrete Wavelet Transform and Daubechies wavelets. Then, a multiresolution decomposition is illustrated for two distinct log-returns series. Finally, a wavelet-based prediction approach is presented, as well as a comparison between its results and those of a traditional prediction method.

Keywords: Wavelet analysis, Discrete Wavelet Transform, financial time series, multiresolution decomposition, prediction.

Resumen

Los métodos wavelet poseen algunas características que los hacen una herramienta con gran potencial para la investigación financiera. El propósito de esta tesis es estudiar la utilidad que tienen los métodos wavelet en el análisis de series de tiempo financieras, para lo cual se han utilizado datos del mercado financiero colombiano.

En esta tesis se presenta brevemente la teoría wavelet, con especial enfoque en la Transformada Discreta Wavelet y en las wavelets de Daubechies. Luego, se ilustra una descomposición multirresolución para dos series diferentes de log-retornos. Finalmente, se presenta un método de predicción basado en wavelets, así como una comparación entre sus resultados y los de un método de predicción tradicional.

Palabras clave: Análisis wavelet, Transformada Discreta Wavelet, serie de tiempo financiera, descomposición multirresolución, predicción.

Contents

List of figures

List of tables

Notations

$\Psi(t)$	Wavelet function.
$\Psi_{a,b}(t)$	Wavelet function at scale a and location b.
$\Psi_{m,n}(t)$	Discretized wavelet function.
$\Psi_{0,0}(t)$	Mother wavelet function.
$x(t)$	Signal.
t	Time variable.
N	Length of the discrete input signal.
M	Number of iterations of wavelet decomposition.
a	Dilation parameter.
b	Translation parameter.
m	Discrete dilation parameter.
n	Discrete translation parameter.
a_o	Fixed dilation step parameter.
b_o	Fixed location step parameter.
C_ψ	Wavelet admissibility condition.
$\widehat{\Psi}(f)$	Fourier transform of the wavelet function.
ι	Imaginary number.
E	Fourier energy spectrum.
$T_{m,n}$	Wavelet (or detail) coefficient.
$\phi(t)$	Scaling function.
$\phi_{m,n}(t)$	Discretized scaling function.
$\phi_{0,0}(t)$	Father wavelet function.
$S_{m,n}$	Approximation coefficient.
$x_m(t)$	Approximation of the signal at scale index m.
$d_m(t)$	Signal detail at scale index m.
c_k	Scaling coefficient.
b_k	Reconfigured version of the scaling coefficients.
N_k	Number of scaling coefficients.
W^m	Wavelet transform vector at scale index m.
r_t	Log-return at time t.
p_t	Price at time t.

Introduction

A wavelet is a function that may be described as a localized wavelike function, and it is used to transform a signal under analysis into another representation which exhibits the signal information in a more useful way. This transformation of a signal through a wavelet function is known as the wavelet transform and from a mathematical perspective may be interpreted as a convolution of the signal with a wavelet function. In a financial time series analysis context, wavelet transform may be viewed as a time-frequency decomposition tool for data analysis.

Likewise, wavelet analysis provides insight into the dynamics of financial time series beyond traditional methodologies. An observed financial time series may contain several structures each occurring on different time scales, and through wavelet methods these structures may be identified and then analyzed with traditional time series tools.

Furthermore, wavelets methods have some additional advantages over other closely related methods; Fourier methods. The main advantage of wavelet methods is that, unlike Fourier methods, have the ability to conserve both time and frequency information which make them a more suitable tool for financial research. Additionally, wavelet methods provide a natural platform to deal with the time-varying characteristics, present in most real-world financial time series, and thus the stationarity assumption could be avoided.

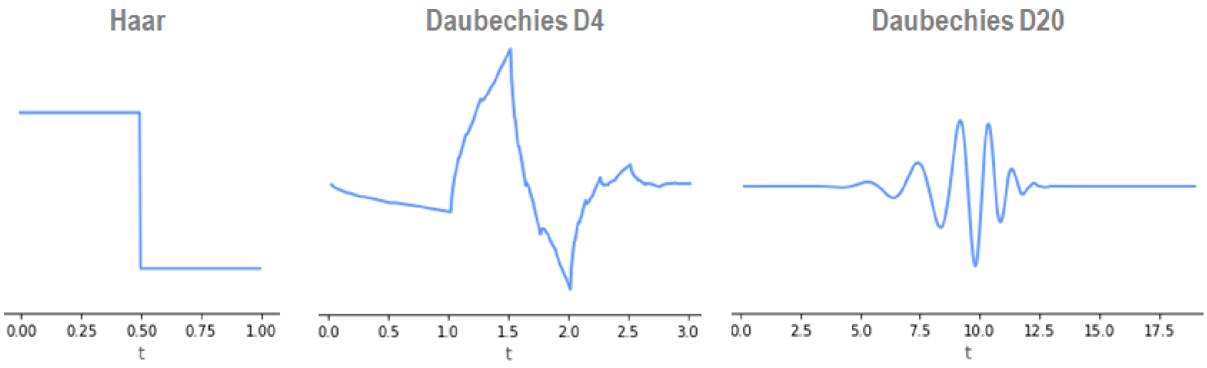
The purpose of this thesis is twofold. First, to introduce briefly the discrete wavelet transform (DWT) theory. Second, to show some of the wavelet analysis benefits, through applications in finance, using Colombian financial market data. Therefore, this thesis has been structured in four main chapters as follow: in the first chapter, the discrete wavelet transform theory is presented including, among others, the scaling function, scaling equation, wavelet equation and the fast wavelet transform; in the second chapter, fundamental characteristics of the Daubechies wavelets family are presented, since these are the type of wavelets used in subsequent chapters; in the third chapter, the wavelet multiresolution decomposition using two different mother wavelets is performed for two distinct kinds of financial assets; in the fourth and final chapter, a wavelet-based approach for financial time series prediction is performed for the two series analyzed in chapter three, which is based on the work of Nguyen and He (2015). In addition, some concluding remarks, ideas for future work and python scripts are given in final sections.

I. The Discrete Wavelet Transform

This chapter covers the basic theory of the discrete wavelet transform (DWT). A wavelet is a function of time $\Psi(t)$, in $L^2(\mathbb{R})$ space, that may be described as a localized wavelike function, and is used to transform a signal $x(t)$ under analysis into another representation which exhibits the signal information in a more useful way. This transformation of a signal through a wavelet function is known as the wavelet transform. As will be shown later, the DWT has some qualities that make it a useful method for analyzing financial time series.

Some examples of wavelet functions which may be used to transform a signal as mentioned above are presented in figure 1. There are two basic ways in which a wavelet can be manipulated: the first one is the translation of the wavelet, and the second one is the scaling of the wavelet. The translation is referred to movements along the time axis and the scaling is referred to the spreading out of the wavelet. These two basic manipulations are used in the discrete wavelet transform of the signal, which means that the DWT is implemented at several locations of the signal, and for several scales of the wavelet, with the purpose to capture features that are local in time and local in frequency.

Figure 1 Some wavelets



It can be seen from the previous figure that wavelets have the form of a small wave, localized on the time axis. All these wavelets (and many others) may be used for analyzing financial data, and should be chosen depending on both the nature of the observed signal of interest and the aim of the analysis. The following are the conditions that a function must satisfy to be a wavelet:

- The admissibility condition:

$$C_\psi = \int_0^\infty \frac{|\widehat{\Psi}(f)|^2}{f} df < \infty \tag{1.1}$$

Where, f is the frequency and $\widehat{\Psi}(f)$ is the Fourier transform of the wavelet function $\Psi(t)$ and is given by:

$$\widehat{\Psi}(f) = \int_{-\infty}^{\infty} \Psi(t) e^{-\iota(2\pi f)t}\, dt \tag{1.2}$$

Where, ι is the imaginary number defined as $\iota = \sqrt{-1}$

- Integration to zero:

  $$\int_{-\infty}^{\infty} \Psi(t)dt = 0 \tag{1.3}$$

  This condition is equivalent to[1] $\widehat{\Psi}(0) = 0$

- Unit energy[2]:

  $$E = \int_{-\infty}^{\infty} |\Psi(t)|^2 dt = 1 \tag{1.4}$$

The wavelet function at scale a and location b is defined as:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\Psi\left(\frac{t-b}{a}\right) \tag{1.5}$$

Where, a and b are the dilation (or scale) and translation (or shifting) parameters respectively.

If discrete values of parameters a and b are considered, the wavelet function may be rewritten by using a logarithmic discretization of the a scale, and making location b proportional to that a scale. This modification leads to a discretized version of the wavelet function which has the form:

$$\Psi_{m,n}(\mathrm{t}) = \frac{1}{\sqrt{a_0^m}}\Psi\left(\frac{t-nb_o a_0^m}{a_0^m}\right) \tag{1.6}$$

Where,
$m \in \mathbb{Z}$: is the discrete dilation parameter
$n \in \mathbb{Z}$: is the discrete translation parameter
a_o: is a fixed dilation step parameter
b_o: is a fixed translation step parameter

Parameters m and n are contained in the set of integers $\mathbb{Z}$. Parameter a_0 must be greater than 1 because of the stability of $\Psi_{m,n}(t)$ (as $m \to \infty$ then $a_0^m \to 0$ and $\Psi_{m,n}(\mathrm{t}) \to \infty$), and parameter b_0 must be greater than 0, since otherwise no wavelet translations could be performed. In addition, note that the size of the translation steps is proportional to the wavelet scale a_0^m through the relationship $\Delta b = b_0 a_0^m$.

Setting $a_o = 2$ and $b_o = 1$ in equation 1.6, leads to an arrangement known as the dyadic grid, which is a power-of-two arrangement for both the dilation and translation steps, and is given by:

$$\Psi_{m,n}(t) = \frac{1}{\sqrt{2^m}}\Psi\left(\frac{t-n2^m}{2^m}\right) \tag{1.7}$$

Where, $\Psi_{0,0}(t) = \Psi(t)$ is known as the mother wavelet function[3].

[1] This is obtained using the Fourier transform of $\Psi(t)$ defined in equation 1.2 as follow:

$$\widehat{\Psi}(0) = \int_{-\infty}^{\infty} \Psi(t)e^{-\imath 2\pi(0)t}dt = \int_{-\infty}^{\infty} \Psi(t)dt = 0$$

[2] The energy functional is defined as the squared norm integrated over its domain.

[3] The equality $\Psi_{0,0}(t) = \Psi(t)$ is obtain through equation 1.7 as follow:

$$\Psi_{0,0}(t) = \frac{1}{\sqrt{2^0}}\Psi\left(\frac{t-(0)2^0}{2^0}\right) = \Psi(t)$$

The dyadic grid arrangement lends itself to the construction of an orthonormal wavelet basis in the $L^2(\mathbb{R})$ space[4], that is, a set of vectors which are perpendicular to each other and can completely define a signal $x(t)$.

The discrete wavelet transform (DWT) of a discrete signal $x(t)$ using discrete wavelets as defined in (1.7) is given by:

$$T_{m,n} = \sum_{t=0}^{N-1} x(t) \frac{1}{\sqrt{2^m}} \Psi\left(\frac{t-n2^m}{2^m}\right) := \langle x, \Psi_{m,n} \rangle \qquad (1.8)$$

Where,
N: is the length[5] of the discrete signal $x(t)$
$T_{m,n}$: are the wavelet (or detail) coefficients, that is, the discrete wavelet transform values for a scale-location grid of index m,n

The DWT may be viewed as the convolution of the signal $x(t)$ with dilated and translated versions of the mother wavelet, by which the wavelet coefficients $T_{m,n}$ at all scale-location indices m,n can be obtained. In addition, note that the wavelet coefficients may be expressed as the inner product in $L^2(\mathbb{R})$, between the signal and the wavelet function (as it is represented in the right side of equation 1.8).

Likewise, the original signal $x(t)$ can be reconstructed through the wavelet coefficients $T_{m,n}$, using **the inverse discrete wavelet transform** which is defined as:

$$x(t) = \sum_{m=1}^{M} \sum_{n=0}^{2^M-1} T_{m,n}\, \Psi_{m,n}\,(t) \qquad (1.9)$$

Where,
$M = \frac{ln(N)}{ln(2)}$: is the number of iterations that can be computed
$N = 2^M$: is the length of the discrete input signal $x(t)$

Equations (1.8) and (1.9) may be described as a decomposition-reconstruction process summarized as follow:

$$\textit{Decomposition process: } x(t) \longrightarrow \langle x, \Psi_{m,n} \rangle \longrightarrow T_{m,n}$$
$$\textit{Reconstruction process: } x(t) \longleftarrow \sum_{m=1}^{M} \sum_{n=0}^{2^M-1} \langle x, \Psi_{m,n} \rangle\, \Psi_{m,n}\,(t) \longleftarrow T_{m,n}$$

I.I. The scaling function

Orthonormal dyadic discrete wavelets are associated with scaling functions which are related with the smoothing of a signal. The scaling function is defined as:

$$\phi_{m,n}(t) = \frac{1}{\sqrt{2^m}} \phi\left(\frac{t-n2^m}{2^m}\right) \qquad (1.10)$$

[4] Since wavelet functions are square-integrable, they are in the $L^2(\mathbb{R})$ space.
[5] This length must be an integer power of 2, because of the dyadic grid arrangement.

Where, $\phi_{0,0}(t) = \phi(t)$ is known as the father wavelet function[6].

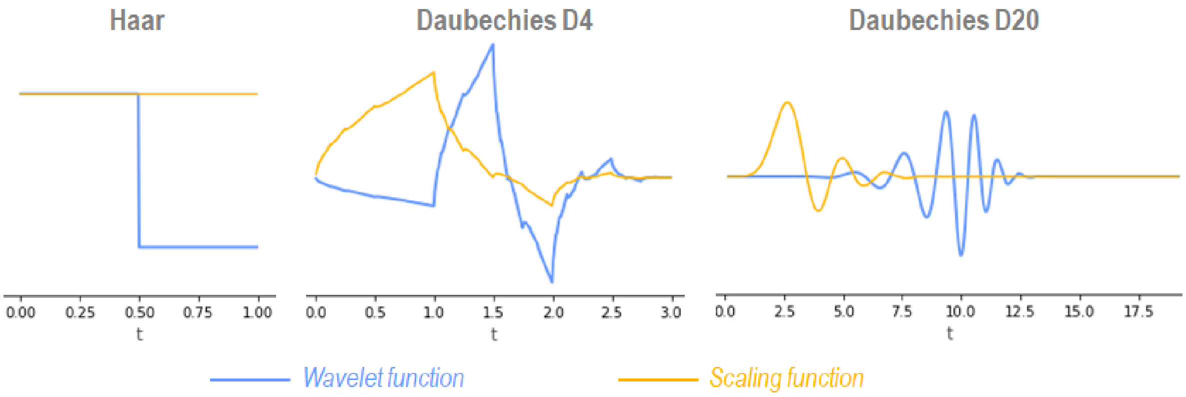
Note that the scaling function (equation 1.10) has the same form as the wavelet function (equation 1.7). As seen later in this chapter, the scaling function is used in the smoothens of the signal, while the wavelet function is used to extract the high signal frequencies.

Furthermore, the scaling function has the following property:

$$\int_{-\infty}^{\infty} \phi_{0,0}(t)\, dt = 1 \tag{1.11}$$

Figure 2 shows the same wavelets presented in figure 1 (blue line), together with their respective scaling functions (yellow line).

Figure 2 Some wavelets and their associated scaling functions



Analogously to the process to obtain the wavelet coefficients $T_{m,n}$ (equation 1.8), the signal $x(t)$ can be convolved with dilated and translated versions of the father wavelet, to obtain the approximation coefficients $S_{m,n}$ at all scale-location indices m,n as follow:

$$S_{m,n} = \sum_{t=0}^{2^M-1} x(t) \frac{1}{\sqrt{2^m}} \phi\left(\frac{t-n2^m}{2^m}\right) := \langle x, \phi_{m,n} \rangle \tag{1.12}$$

Where, $S_{m,n}$ is the approximation coefficient for a scale-location grid of index m,n.

Note that the approximation coefficients may be expressed as the inner product between the signal and the scaling function (right side of equation 1.12). Moreover, the approximation coefficients, at a particular scale index m, are collectively known as the discrete approximation of the signal at scale index m. Likewise, summing a sequence of scaling functions at a particular scale index m, factored by the approximation coefficients, generates an approximation of the signal at scale index m as follow:

$$x_m(t) = \sum_{n=0}^{2^{M-m}-1} S_{m,n}\, \phi_{m,n}(t) \tag{1.13}$$

[6] The equality $\phi_{0,0}(t) = \phi(t)$ is obtain through equation 1.10 as follow:

$$\phi_{0,0}(t) = \frac{1}{\sqrt{2^0}} \phi\left(\frac{t-(0)2^0}{2^0}\right) = \phi(t)$$

Where, $x_m(t)$ is a smooth version of the signal $x(t)$ at scale index m. The smaller the scale index m, the closer the smooth version $x_m(t)$ to the signal $x(t)$.

I.II. The multiresolution representation

The signal $x(t)$ can be represented using both the wavelet and the approximation coefficients (equations 1.9 and 1.13), as shown in the following equation:

$$x(t) = x_{m_0}(t) + \sum_{m=1}^{m_0} d_m(t) \qquad (1.14)$$

Where,
m_0: is an arbitrary scale index
$x_{m_0}(t) = \sum_{n=0}^{2^{M-m_0}-1} S_{m_0,n}\, \phi_{m_0,n}(t)$: is the approximation of the signal at scale index m_0
$d_m(t) = \sum_{n=0}^{2^{M-m}-1} T_{m,n}\, \Psi_{m,n}(t)$: is the signal detail at scale index m

Equation 1.14 leads to the multiresolution representation of the signal, that is, the signal at an arbitrary scale index m expressed as the sum of the signal approximation and the signal detail at the next scale index m+1 as follow:

$$x_m(t) = x_{m+1}(t) + d_{m+1}(t) \qquad (1.15)$$

I.III. The scaling and wavelet equations

The scaling function can be built from contracted and shifted versions of itself through the scaling equation as follow:

$$\phi(t) = \sum_{k=0}^{N_k-1} c_k \phi(2t-k) \qquad (1.16)$$

Where,
$\phi(2t-k)$: is a contracted version of $\phi(t)$ shifted by a step k
c_k: are the scaling coefficients of the scaling function associated to a particular wavelet function of compact support[7]
N_k: is the number of scaling coefficients

In order to ensure an orthogonal system, the scaling coefficients c_k must satisfy the following conditions:

- $\sum_{k=0}^{N_k-1} c_k = 2$ (1.17)
- $\sum_{k=0}^{N_k-1} c_k c_{k+2k'} = \begin{cases} 2 \; if \; k' = 0 \\ 0 \; otherwise \end{cases}$ (1.18)
- $\sum_{k=0}^{N_k-1} (-1)^k c_k k^m = 0$, for integers[8] $m = 0,1,\dots,N_k/2-1$ (1.19)

[7] The word "support" is used to denote a closed set outside of which $\Psi(t) = 0$. Compact support means that this closed set is bounded, and the wavelet function is zero outside a bounded interval. Hence, wavelets functions of compact support have sequences of non-zero scaling coefficients which are of finite length.

[8] These integers m are associated with the moment condition, that is, the wavelets can eliminate parts of the signal which are polynomial up to degree $N_k/2-1$. This condition is required for any of the Daubechies wavelet family.

Similar to the scaling function, the wavelet function can be built from contracted and shifted versions of its scaling function through the wavelet equation as follow:

$$\Psi(t) = \sum_{k=0}^{N_k-1} b_k \phi(2t-k) \qquad (1.20)$$

Where,

$b_k = (-1)^k c_{N_k-1-k}$: are the reconfigured version of the scaling coefficients

In addition, from equations 1.10 and 1.16, a recursive equation for the scaling function can be deduced and is given by:

$$\phi_{m+1,n}(t) = \frac{1}{\sqrt{2}} \sum_{k=0}^{N_k-1} c_k \phi_{m,2n+k}(t) \qquad (1.21)$$

Analogously for the wavelet function, from equations 1.7 and 1.20 a recursive equation can be obtained and is given by:

$$\Psi_{m+1,n}(t) = \frac{1}{\sqrt{2}} \sum_{k=0}^{N_k-1} b_k \phi_{m,2n+k}(t) \qquad (1.22)$$

I.IV. The Fast Wavelet Transform

The approximation and wavelet coefficients may be calculated through recursive equations as explained below, by means of the recursive scaling and wavelet equations presented in the previous section. From the definition of the approximation coefficients given in equation 1.12 and the expression deduced for the scaling function in equation 1.21, the following recursive equation[9] for the approximation coefficients can be found:

$$S_{m+1,n} = \frac{1}{\sqrt{2}} \sum_{k=0}^{N_k-1} c_k S_{m,2n+k} \qquad (1.23)$$

With the initial vector of approximation coefficients given by: $S_{0,n} = x_n = x(t)$.

In a similar way as for the approximation coefficients found in equation 1.20, from equations 1.8 and 1.22 the following recursive equation[10] for the wavelet coefficients can be obtained:

$$T_{m+1,n} = \frac{1}{\sqrt{2}} \sum_{k=0}^{N_k-1} b_k S_{m,2n+k} \qquad (1.24)$$

[9] This equation is obtained by first substituting equation 1.21 in 1.12 as follow:

$$S_{m+1,n} = \sum_{t=0}^{\infty} x(t) \left[\frac{1}{\sqrt{2}} \sum_{k=0}^{N_k-1} c_k \phi_{m,2n+k}(t)\right] = \frac{1}{\sqrt{2}} \sum_{k=0}^{N_k-1} c_k \left[\sum_{t=0}^{\infty} x(t) \phi_{m,2n+k}(t)\right]$$

Finally, note that the expression in brackets is equal to the approximation coefficients at the previous smaller scale and at location $2n+k$, which leads to:

$$S_{m+1,n} = \frac{1}{\sqrt{2}} \sum_{k=0}^{N_k-1} c_k S_{m,2n+k}$$

[10] This equation is obtained by first substituting equation 1.22 in 1.8 as follow:

$$T_{m+1,n} = \sum_{t=0}^{\infty} x(t) \left[\frac{1}{\sqrt{2}} \sum_{k=0}^{N_k-1} b_k \phi_{m,2n+k}(t)\right] = \frac{1}{\sqrt{2}} \sum_{k=0}^{N_k-1} b_k \left[\sum_{t=0}^{\infty} x(t) \phi_{m,2n+k}(t)\right]$$

Finally, note that the expression in brackets is equal to the approximation coefficients at the previous smaller scale and at location $2n+k$, which leads to:

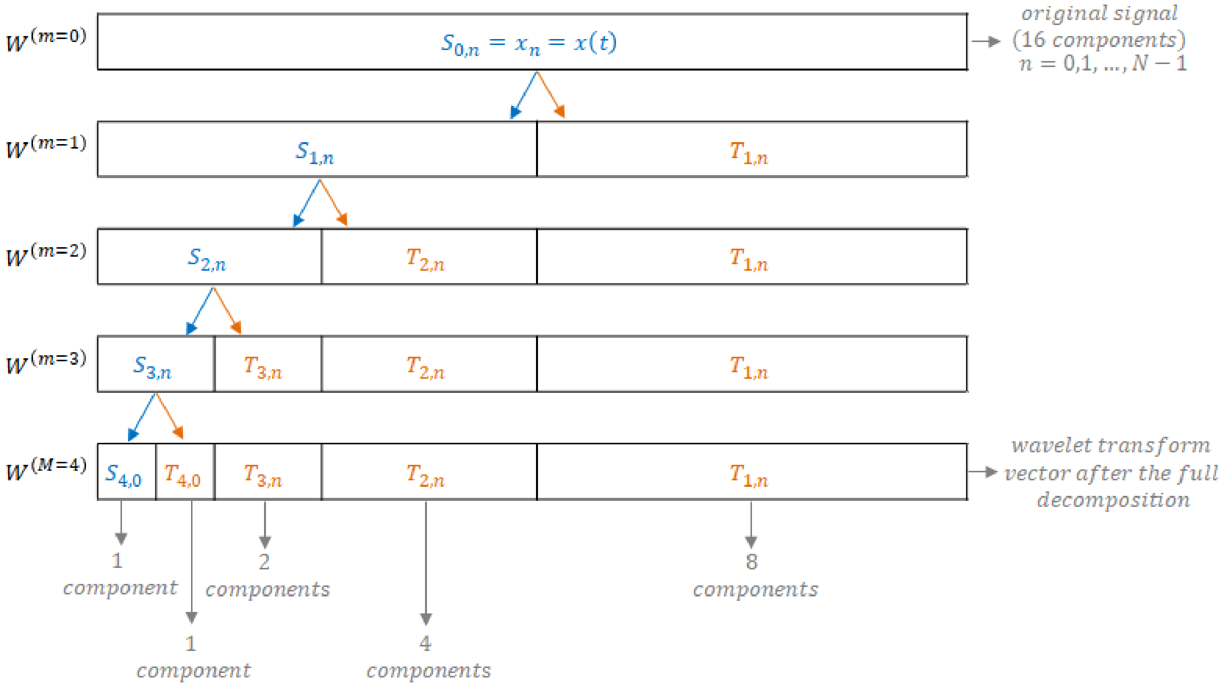
$$T_{m+1,n} = \frac{1}{\sqrt{2}} \sum_{k=0}^{N_k-1} b_k S_{m,2n+k}$$

With the initial vector of approximation coefficients given by: $S_{0,n} = x_n = x(t)$.

Equations 1.23 and 1.24 represent the multiresolution decomposition algorithm and constitute the first half of the fast wavelet transform (FWT).

Furthermore, note that from equations 1.23 and 1.24 is evident that approximation coefficients $S_{m_0,n}$ are the only requirement to calculate the approximation and wavelet coefficients at all subsequent scales to m_0. Figure 3 illustrates this process for an arbitrary input signal vector of length $N = 16$ which implies that $M = 4$ iterations[11] can be performed. After the full decomposition, the wavelet transform vector has the form $W^{(M=4)} = \left(S_{4,0}, T_{4,0}, T_{3,n}, T_{2,n}, T_{1,n}\right)$ which has 16 components as can be seen at the bottom of figure 3, like the original signal[12] and all the other transform vectors $W^{(m)}$ at scale indices $m = 1,2,3$.

Figure 3 Illustration of the multiresolution decomposition algorithm



Likewise, when iterating equations 1.23 and 1.24 to perform multiresolution decomposition algorithm, a highpass and lowpass filtering process[13] of the inputs $S_{m,2n+k}$ to generate the outputs $S_{m+1,n}$ and $T_{m+1,n}$ is implemented. The highpass filter is given by the vector

[11] The number of iterations is obtained as follow:

$$M = \frac{ln(N)}{ln(2)} = \frac{ln(16)}{ln(2)} = 4$$

[12] The original input signal is the transform vector at scale index zero $W^{(m=0)}$.

[13] The highpass filter lets through the high signal frequencies corresponding to the signal details $d_m(t)$, while the lowpass filter lets through the low signal frequencies and hence a smoothed version of the signal $x_m(t)$.

containing the sequences $\frac{1}{\sqrt{2}}b_k$ in equation 1.24, and the lowpass filter is given by the vector containing the sequences $\frac{1}{\sqrt{2}}c_k$ in equation 1.23.

The second half of the fast wavelet transform is the reconstruction algorithm and is defined as:

$$S_{m-1,n} = \frac{1}{\sqrt{2}}\left[\sum_{k=0}^{2^{M-m-1}-1} c_{n-2k}S_{m,k} + \sum_{k=0}^{2^{M-m-1}-1} b_{n-2k}T_{m,k}\right] \tag{1.25}$$

Hence, the approximation coefficients at scale index $m-1$ can be calculated from the approximation and wavelet coefficients at the next scale index m. Moreover, note that the scaling coefficients c_{n-2k} and the reconfigured version of the scaling coefficients b_{n-2k} have non-zero values only when in the range $[0, \dots, N_k - 1]$.

II. Haar and Daubechies wavelets

Since the Haar and Daubechies wavelets are the type of wavelets used in subsequent chapters, some fundamental characteristics of the Daubechies wavelets are presented in this chapter, including the Haar wavelet which is, as shown below, the simplest of the Daubechies wavelets.

II.I. Haar wavelet

The Haar wavelet is the simplest orthonormal wavelet and has only two non-zero scaling coefficients ($N_k = 2$) $c_0 = c_1 = 1$, obtained by solving simultaneously equations 1.17 and 1.18. As seen in previous chapter, the wavelet function can be built from contracted and shifted versions of its scaling function through the wavelet equation (given in equation 1.20). Hence, before introducing the Haar wavelet, the Haar scaling equation is presented in the following, which is obtained by substituting its two scaling coefficients into equation 1.16 as follow:

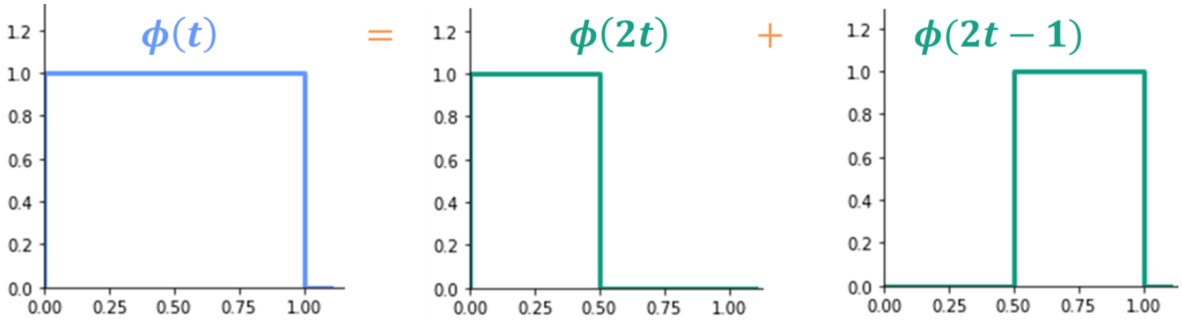
$$\phi(t) = \phi(2t) + \phi(2t-1) \tag{2.1}$$

And its solution leads to the Haar scaling function which is given by:

$$\phi(t) = \begin{cases} 1 & 0 \leq t < 1 \\ 0 & elsewhere \end{cases} \tag{2.2}$$

Figure 4 illustrates the Haar scaling function in terms of contracted and shifted versions of itself, that is, its scaling equation given by equation 2.1.

Figure 4 Haar scaling function



In a similar way to the scaling equation, the Haar wavelet equation is obtained by substituting its two scaling coefficients into equation 1.20 as follow:

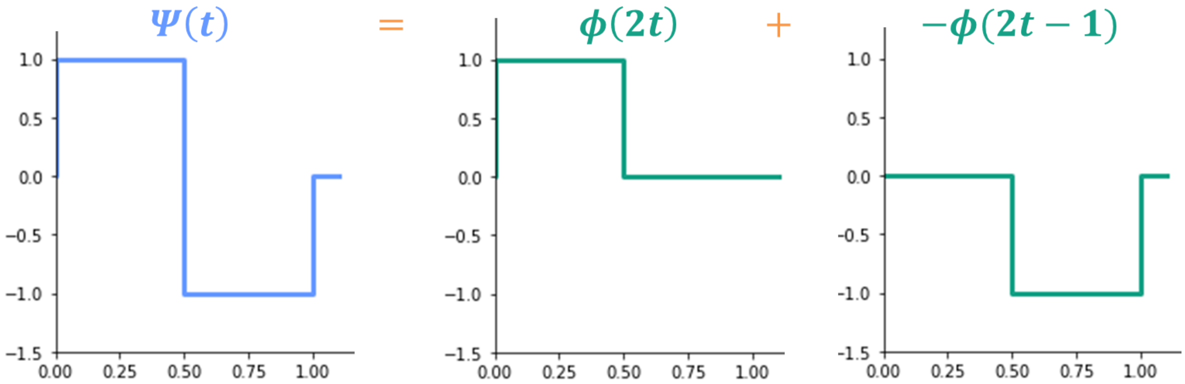
$$\Psi(t) = \phi(2t) - \phi(2t-1) \tag{2.3}$$

And its solution leads to the Haar wavelet function which is given by:

$$\Psi(t) = \begin{cases} 1 & 0 \leq t < 0.5 \\ -1 & 0.5 \leq t < 1 \\ 0 & elsewhere \end{cases} \tag{2.4}$$

Figure 5 illustrates the Haar wavelet function in terms of contracted and shifted versions of its scaling function, that is, its wavelet equation given by equation 2.3.

Figure 5 Haar wavelet function



In addition, note that the Haar wavelet has compact support because of it has a finite number of scaling coefficients $N_k = 2$ and, therefore, a support length of 1. The support length of the Haar and the other Daubechies wavelets is given by their number of scaling coefficients minus one, that is: $support\ length = N_k - 1$.

II.II. Daubechies wavelets

As it can be seen from the previous section, the Haar is a very simple wavelet in the sense that only has two scaling coefficients and both are equal to unity. This section introduces a family of discrete wavelets whose simplest member is the Haar wavelet. These wavelets are known as Daubechies wavelets and, as shown below, satisfy conditions given by equations 1.17 to 1.19.

Likewise, Daubechies wavelets have $N_k/2$ vanishing moments which means that can eliminate segments of the signal which are polynomial up to degree $N_k/2 - 1$ and hence are very good representing polynomial behavior within the signal. In other words, the more scaling coefficients N_k the wavelet has, the higher the number of its vanishing moments $N_k/2$ and hence the higher the degree $N_k/2 - 1$ of polynomial it can eliminate. However, the more scaling coefficients N_k the wavelet has, the larger its support length $N_k - 1$, and hence, the less localized it becomes in the time domain, which makes it less able to isolate singularities of the signal. This is the trade-off that should be considered when choosing the best wavelet for a particular data analysis (see [2] for details).

Daubechies wavelets are usually named with a "D" followed by the number of non-zero scaling coefficients N_k it has, for example, the D10 wavelet is a Daubechies wavelet with $N_k = 10$ non-zero scaling coefficients, the D20 wavelet is a Daubechies wavelet with $N_k =$

20 non-zero scaling coefficients and so on. The calculation for Daubechies scaling coefficients is not trivial; henceforth these are obtained by numerical methods. Table 1 contains the values of the scaling coefficients for Daubechies wavelets up to D20.

Table 1 Daubechies scaling coefficients up to D20

c_k	D2 (Haar)	D4	D6	D8	D10	D12	D14	D16	D18	D20
0	1	0.68301	0.47047	0.32580	0.22642	0.15774	0.11010	0.07696	0.05385	0.03772
1	1	1.18301	1.14112	1.01095	0.85394	0.69950	0.56079	0.44247	0.34483	0.26612
2		0.31699	0.65037	0.89220	1.02433	1.06226	1.03115	0.95549	0.85535	0.74558
3		-0.18301	-0.19093	-0.03958	0.19577	0.44583	0.66437	0.82782	0.92955	0.97363
4			-0.12083	-0.26451	-0.34266	-0.31999	-0.20351	-0.02239	0.18837	0.39764
5			0.04982	0.04362	-0.04560	-0.18352	-0.31684	-0.40166	-0.41475	-0.35334
6				0.04650	0.10970	0.13789	0.10085	0.00067	-0.13695	-0.27711
7				-0.01499	-0.00883	0.03892	0.11400	0.18208	0.21007	0.18013
8					-0.01779	-0.04466	-0.05378	-0.02456	0.04345	0.13160
9					0.00472	0.00078	-0.02344	-0.06235	-0.09565	-0.10097
10						0.00676	0.01775	0.01977	0.00035	-0.04166
11						-0.00152	0.00061	0.01237	0.03162	0.04697
12							-0.00255	-0.00689	-0.00668	0.00510
13							0.00050	-0.00055	-0.00605	-0.01518
14								0.00096	0.00261	0.00197
15								-0.00017	0.00033	0.00282
16									-0.00036	-0.00097
17									0.00006	-0.00016
18										0.00013
19										-0.00002

In order to verify that these Daubechies scaling coefficients lead to an orthogonal system, next, a test of the conditions given by equations 1.17 to 1.19 is illustrated for the D4 wavelet:

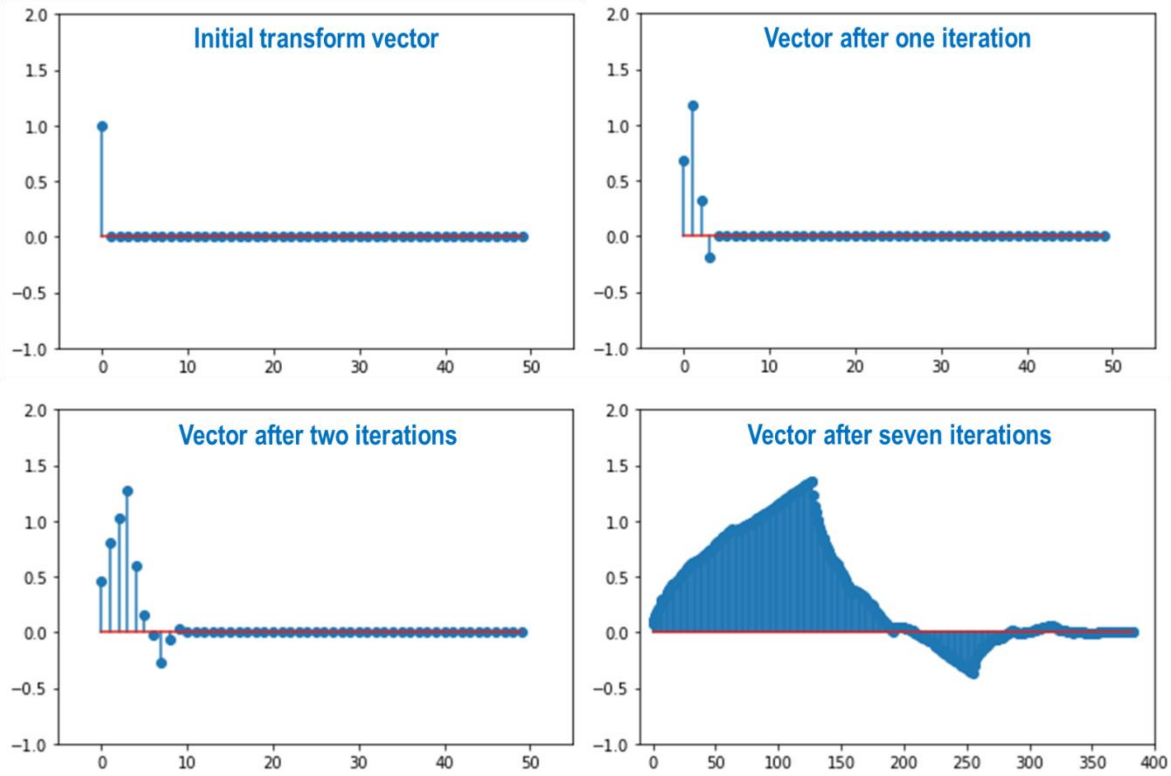
- Test of condition in equation 1.17 for D4 scaling coefficients
$$\begin{aligned}\sum_{k=0}^{N_k-1} c_k &= c_0 + c_1 + c_2 + c_3 \\ &= 0.68301 + 1.18301 + 0.31699 - 0.18301 \\ &= \mathbf{2}\end{aligned}$$

- Test of condition in equation 1.18 for D4 scaling coefficients
$$\begin{aligned}\sum_{k=0}^{N_k-1} c_k^2 &= c_0^2 + c_1^2 + c_2^2 + c_3^2 \\ &= (0.68301)^2 + (1.18301)^2 + (0.31699)^2 + (-0.18301)^2 \\ &= \mathbf{2}\end{aligned}$$

- Test of condition in equation 1.19 for D4 scaling coefficients
  $\sum_{k=0}^{N_k-1}(-1)^k c_k k^m = 0$, for integers $m = 0, N_k/2 - 1 = 1$
  - For $m = 0$
    $\sum_{k=0}^{N_k-1}(-1)^k c_k k^0 = (-1)^0 c_0 0^0 + (-1)^1 c_1 1^0 + (-1)^2 c_2 2^0 + (-1)^3 c_3 3^0$
    $= (1)0.68301 + (-1)1.18301 + (1)0.31699 + (-1)(-0.18301)$
    $= \mathbf{0}$
  - For $m = 1$
    $\sum_{k=0}^{N_k-1}(-1)^k c_k k^1 = (-1)^0 c_0 0^1 + (-1)^1 c_1 1^1 + (-1)^2 c_2 2^1 + (-1)^3 c_3 3^1$
    $= (-1)1.18301(1) + (1)0.31699(2) + (-1)(-0.18301)(3)$
    $= \mathbf{0}$

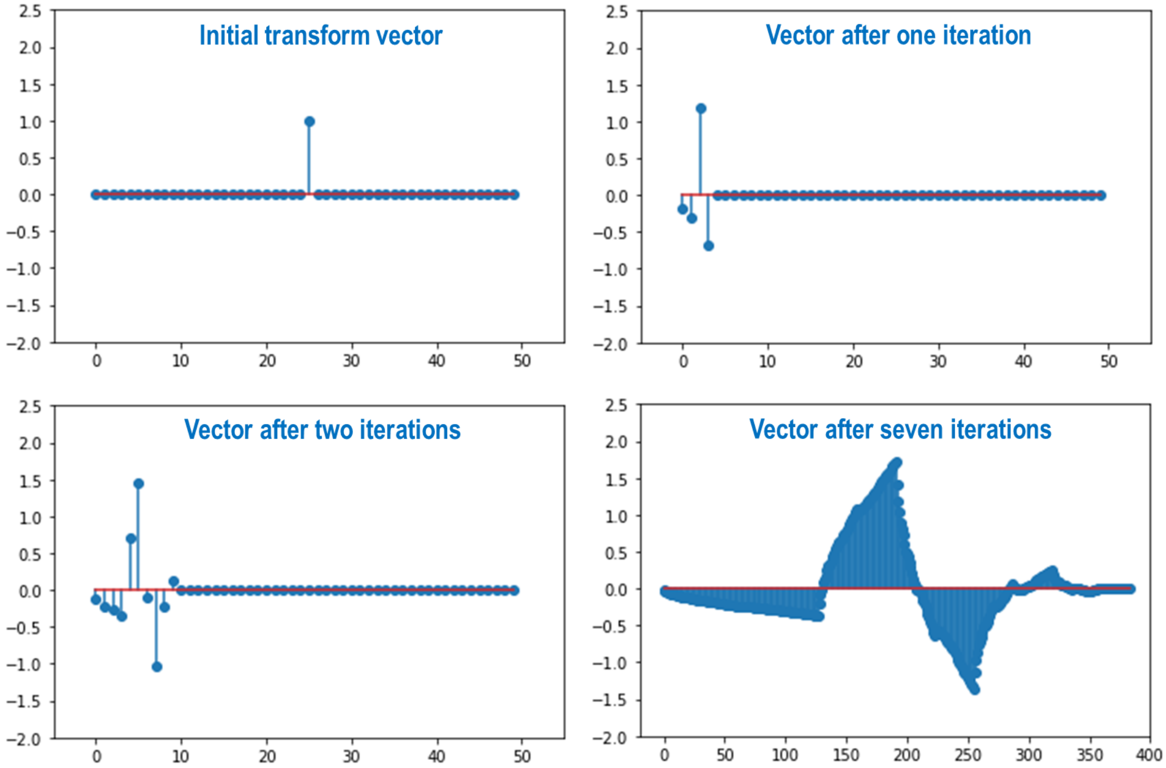
Daubechies scaling and wavelet functions have no explicit formula except for the D2 (Haar) as seen in the previous section. However, a discrete approximation of Daubechies scaling and wavelet functions can be generated through the reconstruction algorithm (equation 1.25). For the scaling function, the approximation can be generated by setting all the values of the transform vector $W^{(M)}$ to zero except the first scaling coefficient $S_{M,0}$ which is set to unity, and then, passing this vector repeatedly through the reconstruction algorithm. This is illustrated in figure 6 which shows, respectively, the initial transform vector $W^{(M)} = [S_{M,0} = 1,0,0,\ldots,0]$, vector after one iteration, vector after two iterations and vector after seven iterations which corresponds to the highest resolution discrete approximation within the presented in the figure for the D4 scaling function.

Figure 6 Discrete approximation of the D4 scaling function



In a similar way, the approximation of the wavelet function can be generated by setting all the values of the transform vector $W^{(M)}$ to zero except the first detail coefficient $T_{M,0}$ which is set to unity, and then, passing this vector repeatedly through the reconstruction algorithm. This is illustrated in figure 7 which shows, respectively, the initial transform vector $W^{(M)} = [0,0, \dots, T_{M,0} = 1,0, \dots ,0]$, vector after one iteration, vector after two iterations and vector after seven iterations which corresponds to the highest resolution discrete approximation within the presented in the figure for the D4 wavelet function.

Figure 7 Discrete approximation of the D4 wavelet function



Note from figures 6 and 7 that the more iterations used, the higher the resolution and hence the better the approximation of the Daubechies scaling and wavelet functions.

III. Wavelet multiresolution decomposition of financial time series

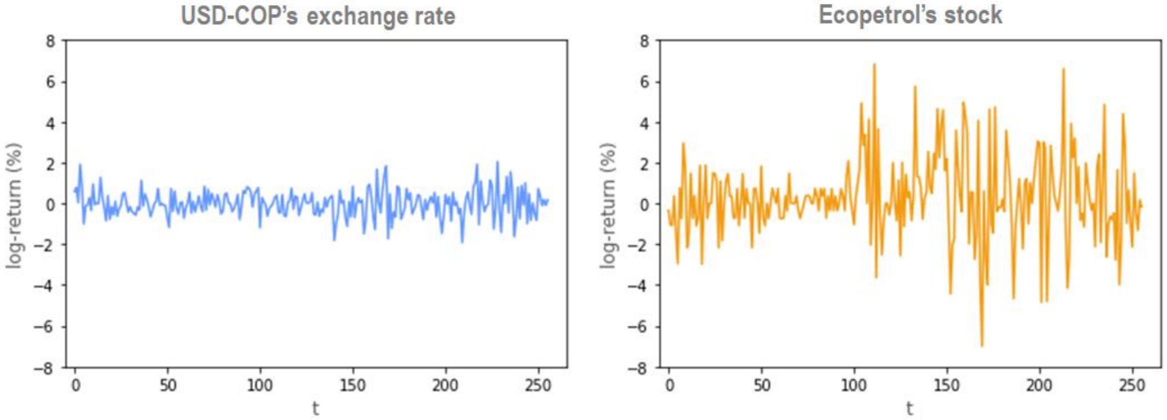
In this chapter, two distinct kinds of financial assets are considered for the wavelet multiresolution decomposition analysis: USD-COP's exchange rate[14] and Ecopetrol's stock. Likewise, the Haar and D4 wavelets are used to decompose each one of these series

As seen in chapter one, equations 1.23 and 1.24 represent the multiresolution decomposition algorithm (first half of the fast wavelet transform) and equation 1.25 represents the reconstruction algorithm (second half of the fast wavelet transform). Hence, these set of equations are employed to calculate the wavelet transform vector W at each scale index m and to reconstruct the original signal from the wavelet transform vector after the full decomposition (i.e. W^M). Likewise, the approximation of the signal $x_m(t)$ and the signal detail $d_m(t)$ are computed for each scale index m through the multiresolution representation of the signal given by equation 1.14.

III.I. Multiresolution decomposition using a Haar wavelet

As mention above, two distinct discrete input signals are considered for the wavelet multiresolution decomposition analysis: USD-COP's exchange rate and Ecopetrol's stock daily log-returns[15] r_t. The length of the two log-returns series is $N = 256$ which implies that $M = 8$ iterations[16] can be performed. The period for USD-COP's exchange rate goes from June 15, 2017 to June 14, 2018 and for Ecopetrol's stock goes from May 23, 2017 to June 14, 2018.

Figure 8 USD-COP's exchange rate and Ecopetrol's stock daily log-returns



[14] The series of this asset was downloaded from Bloomberg.

[15] Log return r_t at time t is calculated trough the following formula:

$$r_t = ln\left(\frac{p_{t+1}}{p_t}\right), \text{ where, } p_t \text{ is the price at time } t.$$

[16] The number of iterations is obtained as follow:

$$M = \frac{ln(N)}{ln(2)} = \frac{ln(256)}{ln(2)} = 8$$

Figure 8 shows the USD-COP's exchange rate (blue line) and Ecopetrol's stock (orange line) daily log-returns series. From these two charts, an evident difference in the behavior of the two assets under analysis is observed. The variance in the returns for USD-COP's exchange rate remains at a relatively low level, while for Ecopetrol's stock there is an important increase in returns variance as from the second third of the series; the variance for USD-COP's exchange rate is 0.4338 while for Ecopetrol's stock is 4.0507 for the full series, 1.1860 for the first third and 5.4389 for the last two thirds of the series.

Before implementing the fast wavelet transform (FWT) note that since the Haar wavelet has only two scaling coefficients and both are equal to unity, the recursive equation for the approximation coefficients (equation 1.23) becomes:

$$S_{m+1,n} = \frac{1}{\sqrt{2}}\left(S_{m,2n} + S_{m,2n+1}\right) \tag{3.1}$$

Analogously, the recursive equation for the wavelet coefficients (equation 1.24) becomes:

$$T_{m+1,n} = \frac{1}{\sqrt{2}}\left(S_{m,2n} - S_{m,2n+1}\right) \tag{3.2}$$

So, through equations 3.1 and 3.2 the Haar wavelet decomposition can be performed. Let y_n be the vector containing the Ecopetrol's stock daily log-returns series and z_n be the vector containing the USD-COP's exchange rate daily log-returns series. Likewise, as seen in chapter one, to calculate the approximation and wavelet coefficients for $m > 0$, it is necessary to know the values of the approximation coefficients at scale $m = 0$, that is, y_n and z_n respectively. In the following it is illustrated how to calculate the first two approximation and wavelet coefficients at scale index $m = 1$ for Ecopetrol's stock daily log-returns series (i.e. y_n):

$$S_{1,0} = \frac{1}{\sqrt{2}}\left(S_{0,0} + S_{0,1}\right) = \frac{1}{\sqrt{2}}(-0.3514 - 1.0619) = -0.9994 \tag{3.3}$$

$$S_{1,1} = \frac{1}{\sqrt{2}}\left(S_{0,2} + S_{0,3}\right) = \frac{1}{\sqrt{2}}(-1.0733 + 0.3590) = -0.5050 \tag{3.4}$$

$$T_{1,0} = \frac{1}{\sqrt{2}}\left(S_{0,0} - S_{0,1}\right) = \frac{1}{\sqrt{2}}(-0.3514 + 1.0619) = 0.5023 \tag{3.5}$$

$$T_{1,1} = \frac{1}{\sqrt{2}}\left(S_{0,2} - S_{0,3}\right) = \frac{1}{\sqrt{2}}(-1.0733 - 0.3590) = -1.0128 \tag{3.6}$$

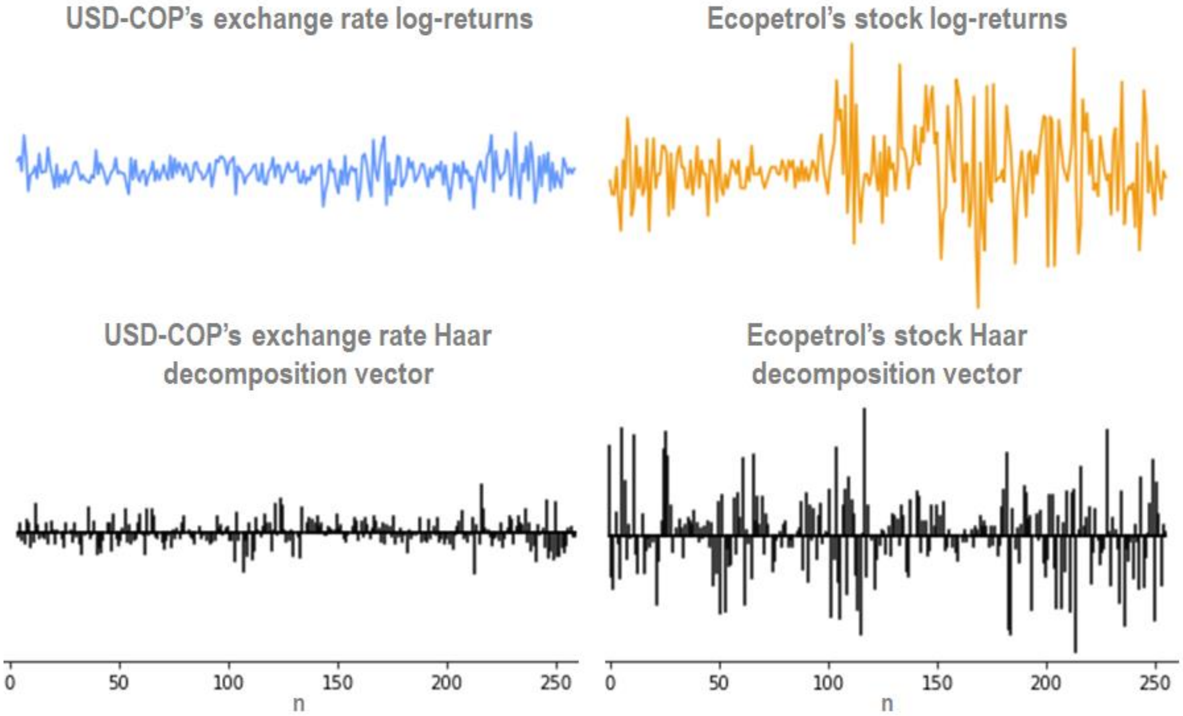
So, the transform vector after the first iteration is given by:

$$\begin{aligned} W^{(m=1)} &= \left[S_{1,0}, S_{1,1}, \dots, S_{1,127}, T_{1,0}, T_{1,0}, \dots, T_{1,127}\right] \\ &= \left[-0.9994, -0.5050, \dots, S_{1,127}, 0.5023, -1.0128, \dots, T_{1,127}\right] \end{aligned} \tag{3.7}$$

The Haar decomposition vector (i.e. the vector after the full decomposition, $W^{(M=8)}$) of the USD-COP's and Ecopetrol log-returns series are shown in the lower part of figure 9 and, to facilitate comparison, the vertical axis of each chart in figure is exactly at the same scale[17].

[17] The range of the vertical axis of each chart corresponds to the interval $[-10\%, 10\%]$.

Figure 9 Haar decomposition of USD-COP and Ecopetrol log-returns



Each of the two vectors after the full decomposition (i.e. $W^{(M=8)}$) contains 256 components (255 wavelet coefficients and a signal mean coefficient). Table 2 illustrates the composition of the two Haar decomposition vectors shown in figure 9, which can be deduced from figure 3.

Table 2 Structure of the Haar decomposition vectors shown in figure 9

Sub-vectors of $\boldsymbol{W^{(M=8)}}$	sub-vector length	Description
$\boldsymbol{S_{8,0}}$	$2^{M-m} = 2^0 = 1$	Approximation coefficient for a scale-location grid of index 8,0 (signal mean coefficient at scale index $M = 8$)
$\boldsymbol{T_{8,0}}$	$2^{M-m} = 2^0 = 1$	Wavelet coefficient for a scale-location grid of index 8,0
$\boldsymbol{T_{7,n}}$	$2^{M-m} = 2^1 = 2$	Wavelet coefficients for a scale-location grid of index 7,n
$\boldsymbol{T_{6,n}}$	$2^{M-m} = 2^2 = 4$	Wavelet coefficients for a scale-location grid of index 6,n
$\boldsymbol{T_{5,n}}$	$2^{M-m} = 2^3 = 8$	Wavelet coefficients for a scale-location grid of index 5,n
$\boldsymbol{T_{4,n}}$	$2^{M-m} = 2^4 = 16$	Wavelet coefficients for a scale-location grid of index 4,n
$\boldsymbol{T_{3,n}}$	$2^{M-m} = 2^5 = 32$	Wavelet coefficients for a scale-location grid of index 3,n
$\boldsymbol{T_{2,n}}$	$2^{M-m} = 2^6 = 64$	Wavelet coefficients for a scale-location grid of index 2,n
$\boldsymbol{T_{1,n}}$	$2^{M-m} = 2^7 = 128$	Wavelet coefficients for a scale-location grid of index 1,n

As seen in chapter one, the reconstruction algorithm (second half of the fast wavelet transform) given by equation (1.25) is used to bring back the original discrete input signals y_n and z_n from the Haar decomposition vectors. In the following is illustrated how to return from the approximation $S_{1,n}$ and wavelet $T_{1,n}$ coefficients at scale index $m = 1$ (obtained in equation 3.7), to the first four observations of the original signal input y_n (i.e. Ecopetrol's stock daily log-returns series):

$$\begin{aligned} S_{0,0} &= \frac{1}{\sqrt{2}}\left[\textstyle\sum_{k=0}^{2^7-1} c_{0-2k} S_{1,k} + \sum_{k=0}^{2^7-1} b_{0-2k} T_{1,k}\right] \\ &= \frac{1}{\sqrt{2}}\left[S_{1,0} + T_{1,0}\right] = \frac{1}{\sqrt{2}}[-0.9995 + 0.524] = -0.3514 \qquad (3.8) \\ S_{0,1} &= \frac{1}{\sqrt{2}}\left[\textstyle\sum_{k=0}^{2^7-1} c_{1-2k} S_{1,k} + \sum_{k=0}^{2^7-1} b_{1-2k} T_{1,k}\right] \\ &= \frac{1}{\sqrt{2}}\left[S_{1,0} + (-1)T_{1,0}\right] = \frac{1}{\sqrt{2}}[-0.9995 - 0.524] = -1.0619 \qquad (3.9) \\ S_{0,2} &= \frac{1}{\sqrt{2}}\left[\textstyle\sum_{k=0}^{2^7-1} c_{2-2k} S_{1,k} + \sum_{k=0}^{2^7-1} b_{2-2k} T_{1,k}\right] \\ &= \frac{1}{\sqrt{2}}\left[S_{1,1} + T_{1,1}\right] = \frac{1}{\sqrt{2}}[-0.5051 - 1.0129] = -1.0733 \qquad (3.10) \\ S_{0,3} &= \frac{1}{\sqrt{2}}\left[\textstyle\sum_{k=0}^{2^7-1} c_{3-2k} S_{1,k} + \sum_{k=0}^{2^7-1} b_{3-2k} T_{1,k}\right] \\ &= \frac{1}{\sqrt{2}}\left[S_{1,1} + (-1)T_{1,1}\right] = \frac{1}{\sqrt{2}}[-0.5051 + 1.0129] = 0.3590 \qquad (3.11) \end{aligned}$$

So, in this way and through the second half of the fast wavelet transform the original signal inputs y_n and z_n can be completely recovered without loss of information. Likewise, from the Haar decomposition vectors at each scale m, the corresponding approximations $x_m(t)$ and details $d_m(t)$ of the signal can be constructed for each scale m through the multiresolution representation of the signal (equation 1.14) and the Haar scaling (equation 2.2) and wavelet (equation 2.4) functions.

Figure 10 Haar multiresolution decomposition of the USD-COP's log-returns series

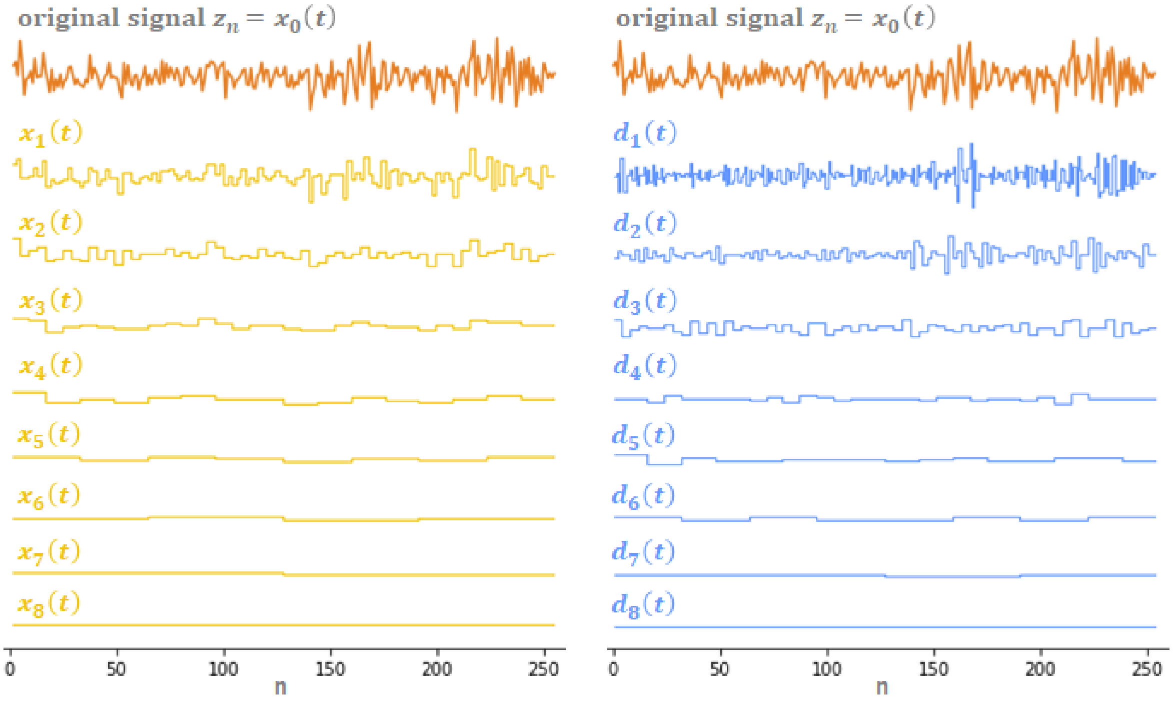


Figure 10 shows the original signal z_n (at the top) and the approximations $x_m(t)$ and details $d_m(t)$ of the USD-COP's log-returns series at all scale indices from $m = 1$ to $M =$

8, using a Haar wavelet for the multiresolution decomposition. Adding together all the details from $d_1(t)$ to $d_8(t)$ plus the signal approximation at scale $M = 8$ (which corresponds to signal's mean) results in the original signal. In addition, note from equation 1.15 that the original signal can also be obtained by adding the approximation and detail signal at scale index $m = 1$, that is, $x_0(t) = x_1(t) + d_1(t)$. In fact, the approximations of the signal at any scale index from $m = 1$ to $m = 7$ can be obtained in this way.

Furthermore, remember from chapter one that when performing equation 1.14, a lowpass and highpass filtering process is implemented through the sequences $\frac{1}{\sqrt{2}}c_k$ and $\frac{1}{\sqrt{2}}b_k$. As illustrated in figure 10, the lowpass filter let through the low signal frequencies and hence results in a smoothed version of the signal $x_m(t)$, while the highpass filter let through the high signal frequencies corresponding to the signal details $d_m(t)$. Note that according to the signal details, the largest fluctuations in USD-COP's exchange rate log-returns occur across the first three scales.

Figure 11 Haar multiresolution decomposition of the Ecopetrol's log-returns series

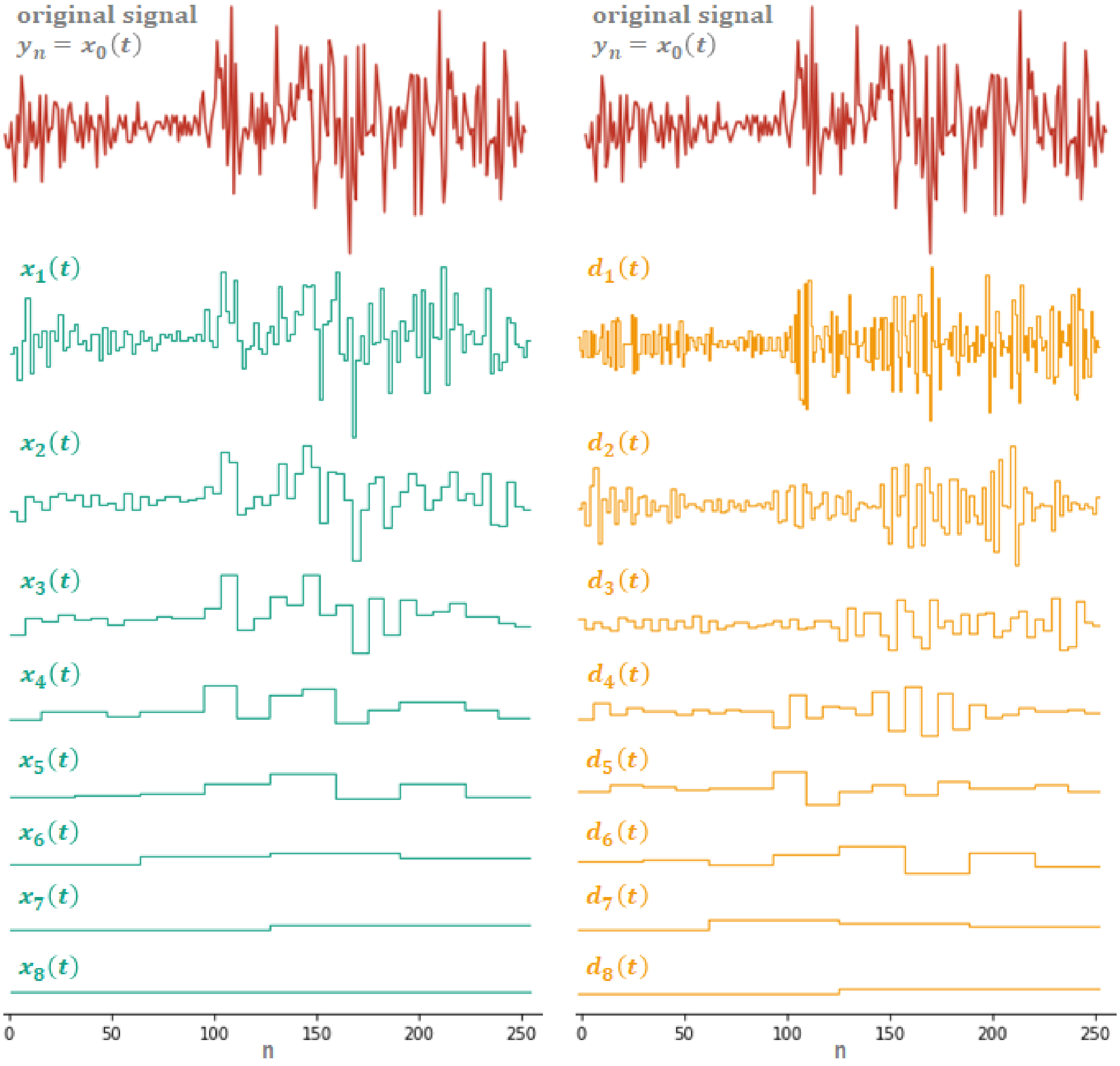
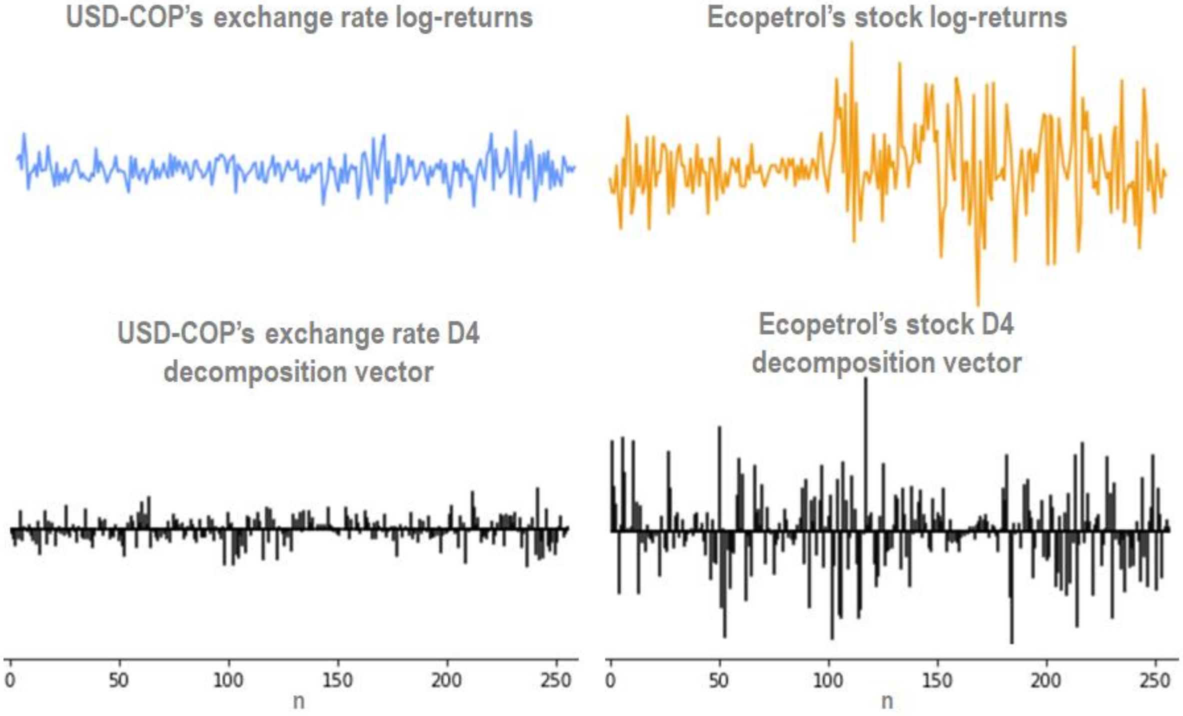


Figure 11 shows the same multiresolution decomposition performed for the USD-COP's log-returns but instead for the Ecopetrol log-returns. The original signal y_n is plot at the top of the figure and below it, the approximations $x_m(t)$ and details $d_m(t)$ of the Ecopetrol log-returns series at all scale indices from $\mathrm{m} = 1$ to $\mathrm{M} = 8$. Two things are noticeable from the figure: the first one is that there is a large group of rapidly fluctuating returns as from the second third of the series and according to the signal details the large fluctuations in Ecopetrol's stock log-returns occur across the first four scales towards the days 100 to 256 (i.e. from August 10, 2017 to June 14, 2018). The second one is that, as expected, the highest frequency oscillations are captured at the smallest scales.

III.II. Multiresolution decomposition using a D4 wavelet

In this section, the same multiresolution decomposition is performed using a D4 wavelet. Since the Daubechies D4 wavelet has more scaling coefficients than the Haar, its support length is larger and hence is less localized in the time domain and thus less able to isolate singularities in the signal. However, as seen in chapter two, Daubechies D4 wavelet is better than the Haar at representing polynomial behavior within the signal because of it has more vanishing moments[18].

Figure 12 D4 wavelet decomposition of USD-COP and Ecopetrol log-returns



[18] The Daubechies D4 has $\frac{N_k}{2} = 2$ vanishing moments which means that can eliminate segments of the signal which are polynomial up to degree $\frac{N_k}{2} - 1 = 1$ versus the Haar wavelet that has just 1 vanishing moment.

The D4 decomposition vectors (i.e. the vectors after the full decomposition $W^{(M=8)}$) of the USD-COP's and Ecopetrol log-returns series are shown in the lower part of figure 12 and, to facilitate comparison, the vertical axis of each chart in figure is exactly at the same scale as in figure 9. Each of the two vectors after the full decomposition (i.e. $W^{(M=8)}$) contains 256 components (255 wavelet coefficients and a signal mean coefficient). Note from this figure that for large oscillations of the signals under analysis, the D4 wavelet coefficients reach larger values than those presented in figure 9 for the Haar wavelet, which means that captures the high frequency oscillations of the signals better than the Haar wavelet as seen in subsequent charts.

Figure 13 D4 multiresolution decomposition of the USD-COP's log-returns series

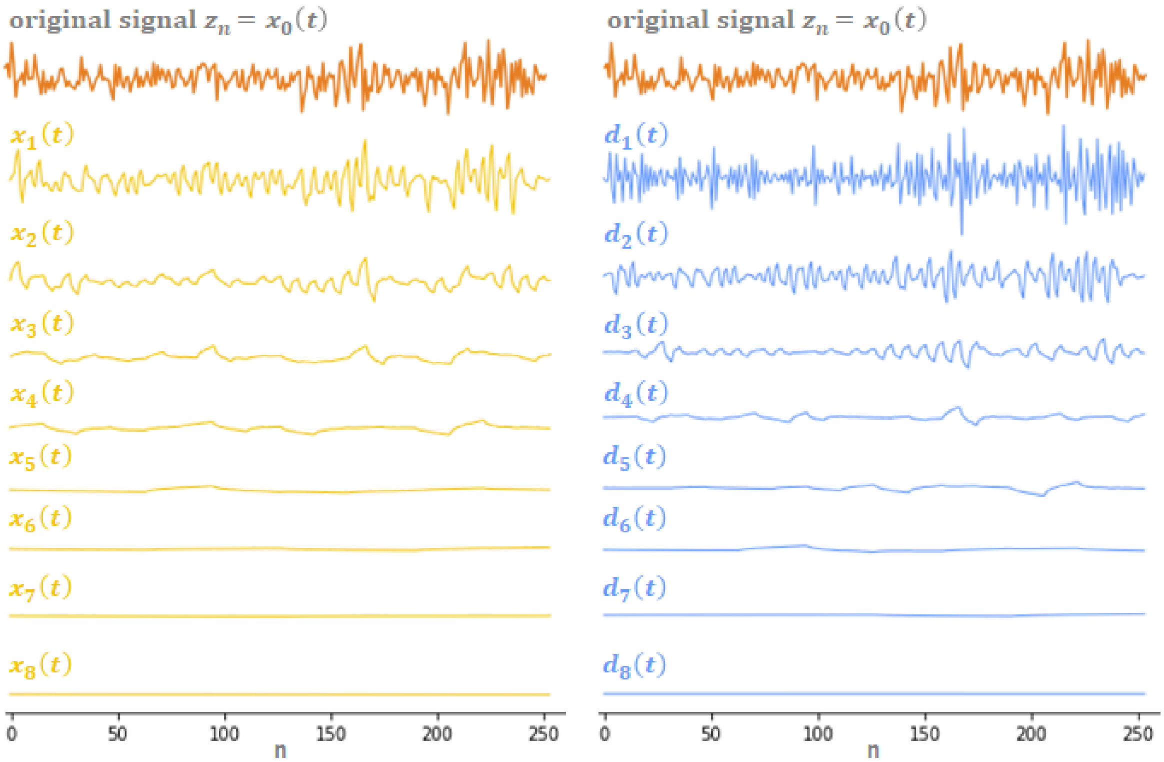
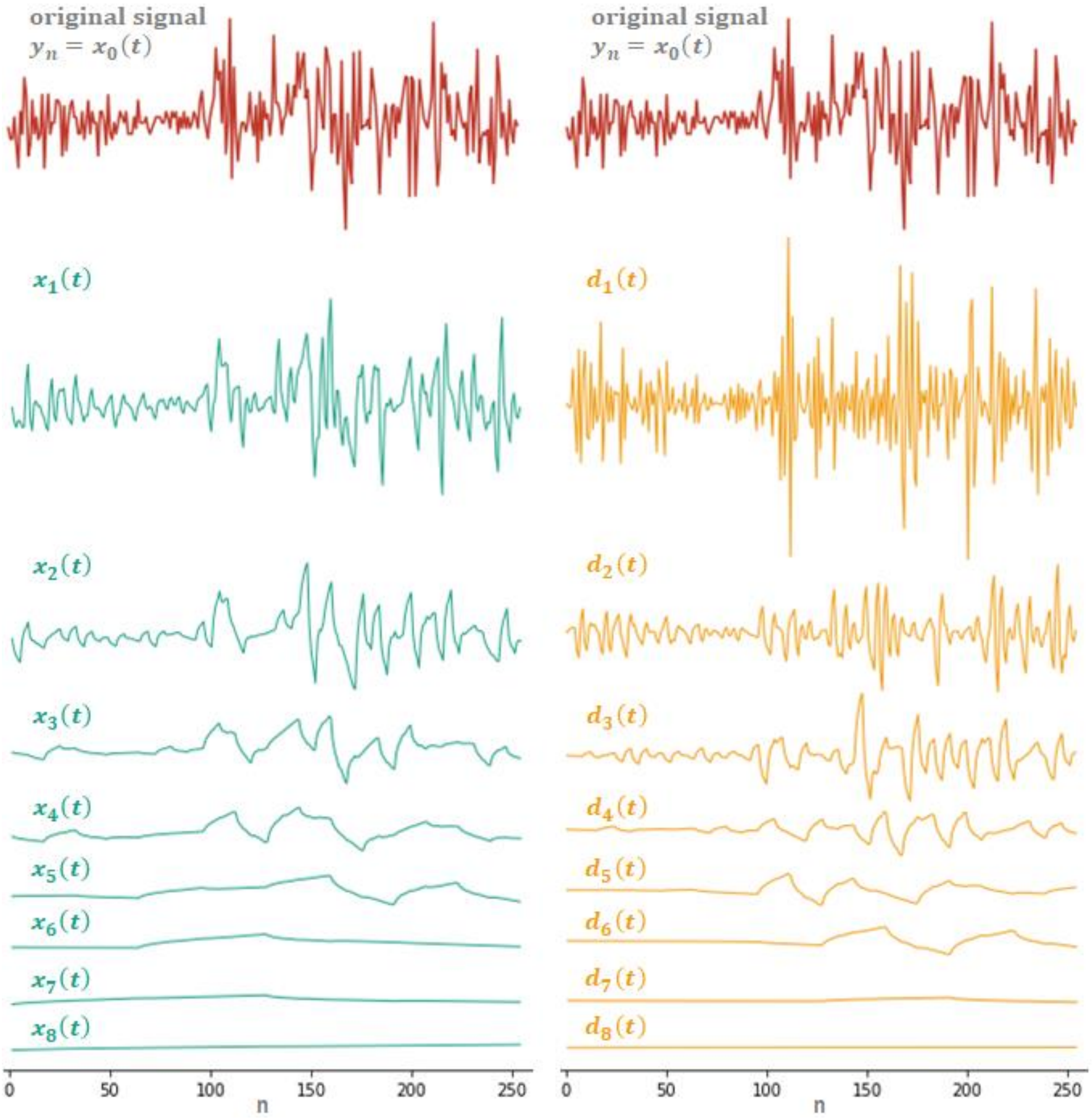


Figure 13 shows the original signal z_n (at the top) and the approximations $x_m(t)$ and details $d_m(t)$ of the USD-COP's log-returns series at all scale indices from $m = 1$ to $M = 8$, using a D4 wavelet for the multiresolution decomposition. The difference in shapes between the charts in this figure and those in figure 10 is explained by the particular mother wavelet used for the decomposition in each case; the approximations $x_m(t)$ and details $d_m(t)$ of the USD-COP's log-returns series at each scale indices from $m = 1$ to $M = 8$ are composed of piecewise wavelet functions (i.e. D4 or Haar wavelet basis functions). Likewise, the signal is smoothed faster using a D4 wavelet than using a Haar wavelet because the first one is relatively smoother. In addition, note also that, as expected, the highest frequency oscillations are captured at the smallest scales and according to the signal details the largest fluctuations in USD-COP's exchange rate log-returns occur across the first three scales.

Figure 14 D4 multiresolution decomposition of the Ecopetrol's log-returns series

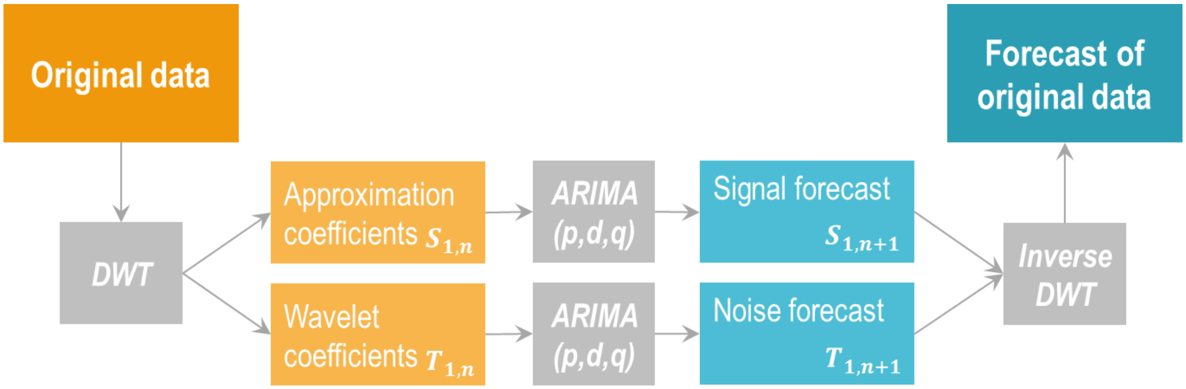


The same D4 multiresolution for Ecopetrol's log-returns is shown in figure 14. The original signal y_n is plotted at the top of the figure and below it, the approximations $x_m(t)$ and details $d_m(t)$ of the Ecopetrol log-returns series at all scale indices from $m = 1$ to $M = 8$. Three things are noticeable from this figure: first, according to the signal details the large fluctuations in Ecopetrol's stock log-returns occur across the first three scales towards the days 100 to 256 (i.e. from August 10, 2017 to June 14, 2018). Second, the different shapes of the charts in this figure versus those in figure 10 are caused by the particular mother wavelet used for the decomposition in each case. Third, the signal is smoothens faster using a D4 wavelet than using a Haar wavelet because the first one is smoother.

IV. Financial time series prediction based on the Discrete Wavelet Transform

In this chapter, a wavelet-based approach for financial time series prediction is performed for both: the USD-COP's exchange rate and the Ecopetrol's stock daily log-returns. The mentioned approach is based on the work of Nguyen and He (2015) and may be summarized by figure 15.

Figure 15 Wavelet-based approach for financial time series prediction


Original data
DWT
Approximation coefficients $S_{1,n}$
Wavelet coefficients $T_{1,n}$
ARIMA (p,d,q)
ARIMA (p,d,q)
Signal forecast $S_{1,n+1}$
Noise forecast $T_{1,n+1}$
Inverse DWT
Forecast of original data

According to this figure, the wavelet-based approach proposed for time series prediction is implemented in three steps (gray boxes). First, performing the discrete wavelet transform to decompose original series to obtain the approximation and wavelet coefficients at the first scale index, that is, $W^{m=1} = [S_{1,0}, \dots, S_{1,n}, T_{1,0}, \dots, T_{1,n}]$. Second, fitting an Autorregresive Integrated Moving Average Model, denoted as ARIMA(p,d,q)[19], for both sub-series (the approximation coefficients $S_{1,n}$ and the wavelet coefficients $T_{1,n}$) to predict one data point forward into the future, that is, to obtain $S_{1,n+1}$ and $T_{1,n+1}$ respectively. Third, applying the inverse discrete wavelet transform to the new vector of approximation and wavelet coefficients (including the forecasted values $S_{1,n+1}$ and $T_{1,n+1}$) to obtain the forecast of the original series for two steps forward into the future[20]. Note that the wavelet-based approach proposed can be performed using another forecasting method instead of an ARIMA(p,d,q) model like, for instance, a neural network.

IV.I Implementing the wavelet-based prediction approach

- **First step: obtain the approximation and wavelet coefficients at the first scale index.** The analyzed period for USD-COP's exchange rate goes from March 16, 2018 to

[19] Parameters p, d and q represent, respectively, the number of lag observations included in the model, the number of times the data have had past values subtracted (degree of differencing) and the size of the moving average window.

[20] Since the length of the new vector of approximation and wavelet coefficients is larger than the length of the original vector (i.e. $W^{m=1}$) in two data points (i.e. $S_{1,n+1}$ and $T_{1,n+1}$), the first approximation and wavelet coefficients (i.e. $S_{1,0}$ and $T_{1,0}$) are eliminated before performing the inverse discrete wavelet transform in order to guarantee the dyadic grid structure, as explained in chapter one.

June 14, 2018 and for Ecopetrol's stock goes from March 8, 2018 to June 14, 2018. Hence, the length of each log-returns series is $N = 64$ which leads, after the first wavelet decomposition iteration (i.e. $W^{m=1}$) to four length $N = 32$ sub-series ($S_{1,n}^{COP-USD}, T_{1,n}^{COP-USD}, S_{1,n}^{Ecopetrol}$ and $T_{1,n}^{Ecopetrol}$). The mother wavelet used in the decomposition was the Daubechies D2 (Haar) wavelet.

- **Second step: predict one data point forward into the future for the four sub-series.** In order to fit an ARIMA(p,d,q) model for each sub-series of the USD-COP's exchange rate and the Ecopetrol's stock daily log-returns (i.e. $S_{1,n}^{COP-USD}, T_{1,n}^{COP-USD}, S_{1,n}^{Ecopetrol}$ and $T_{1,n}^{Ecopetrol}$), in the following is explained the procedure used to obtain the optimal order (p,d,q) for each case. There are several alternatives to obtain that optimal order and for this thesis a two-parts procedure has been followed; the first part consists in evaluating an ARIMA model and the second one in evaluating for different sets of order parameters (p,d,q), and choosing the best one according to the mean squared error (MSE). This was performed through the python's library "statsmodels" as shown in the appendix.

The first part of the procedure (evaluate an ARIMA model) is implemented in three steps. The first step is to split the dataset into two: two thirds for the initial training (i.e. fitting) dataset and the remaining third for the test dataset. The second step consists on making one-step predictions of the test dataset (i.e. the last third of the initial dataset) and stored them. The step three is to calculate the mean squared error (MSE), which is performed from comparing the prediction for each time-step versus its corresponding real value in the test dataset.

The second part of the procedure (evaluate a set of parameters) consists in evaluate, as in the first part of the procedure, ARIMA models with different combinations of parameters (p,d,q) from a predefined set of values for each one. Then, the optimal order (p,d,q) corresponds to the one of the ARIMA model with the lower mean squared error.

Table 3 Optimal ARIMA(p,d,q) model for sub-series $S_{1,n}$ and $T_{1,n}$

	USD-COP's sub-series MSE		Ecopetrol sub-series MSE	
Order (p,d,q)	Approximation $S_{1,n}^{COP-USD}$	Wavelet $T_{1,n}^{COP-USD}$	Approximation $S_{1,n}^{Ecopetrol}$	Wavelet $T_{1,n}^{Ecopetrol}$
(0,0,0)	**0.327**	**0.845**	4.484	5.955
(0,0,1)	0.341	0.901	8.272	5.175
(0,0,2)	0.389	0.979	-	5.320
(0,0,3)	0.403	1.589	-	5.042
(0,0,4)	0.442	0.965	-	7.052
(0,1,0)	0.718	1.732	7.911	18.239
(0,1,1)	0.484	0.897	**4.290**	6.519

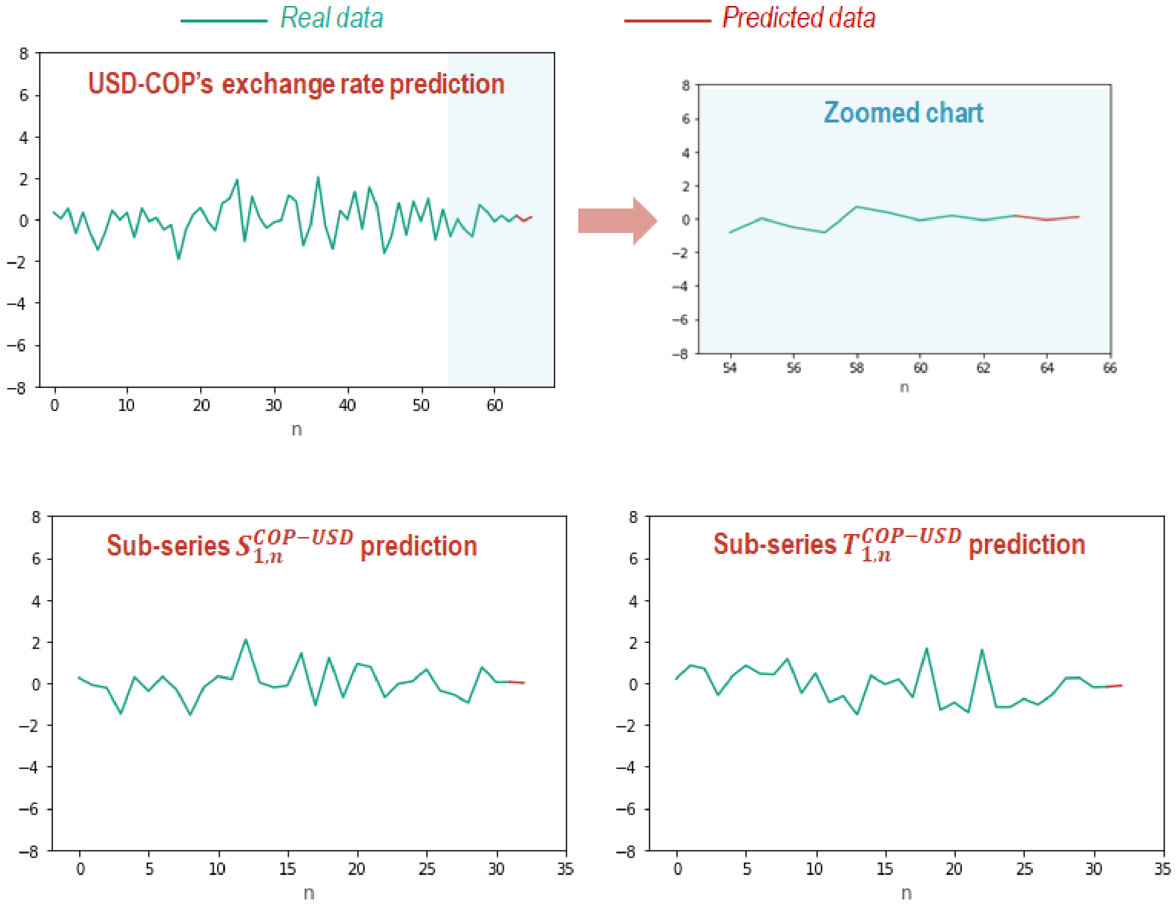
	USD-COP's sub-series MSE		Ecopetrol sub-series MSE	
Order (p,d,q)	Approximation $S_{1,n}^{COP-USD}$	Wavelet $T_{1,n}^{COP-USD}$	Approximation $S_{1,n}^{Ecopetrol}$	Wavelet $T_{1,n}^{Ecopetrol}$
(0,2,0)	2.274	5.331	22.794	65.157
(0,2,1)	0.809	1.912	8.735	19.798
(1,0,0)	0.358	0.921	6.039	5.359
(1,0,1)	-	-	9.409	5.307
(1,1,0)	0.758	1.326	6.789	11.167
(1,1,1)	0.586	0.944	-	-
(1,2,0)	1.774	3.306	12.873	32.595
(2,0,0)	0.413	0.981	7.584	5.192
(2,0,1)	-	-	9.275	5.212
(2,0,2)	-	-	-	**4.399**
(2,1,0)	0.685	1.323	7.928	6.944
(2,1,1)	0.604	1.540	6.318	6.374
(2,2,0)	1.371	3.476	12.149	16.637
(2,2,1)	-	-	9.139	7.839
(3,0,0)	0.425	1.069	8.327	5.242
(3,0,1)	0.431	-	-	5.287
(3,1,0)	0.651	0.944	8.425	6.907
(3,1,1)	0.680	-	6.773	7.101
(3,2,0)	1.365	1.858	-	14.907
(3,2,1)	-	-	9.657	7.719
(4,0,0)	0.436	0.880	8.293	5.563
(4,0,1)	0.436	-	9.003	5.860
(4,1,0)	0.451	1.108	-	6.348
(4,1,1)	0.757	-	5.798	6.847
(5,0,0)	0.423	1.062	8.056	5.953
(5,1,0)	0.647	1.456	-	6.267

Table 3 contains, for each one of the analyzed sub-series (i.e. $S_{1,n}^{COP-USD}, T_{1,n}^{COP-USD}, S_{1,n}^{Ecopetrol}$ *and* $T_{1,n}^{Ecopetrol}$), the mean squared error (MSE) for different combinations of parameters (p,d,q) in the ranges of values $p = [0,1,2,3,4,5], d = [0,1,2]$ *and* $q = [0,1,2,3,4]$, and the optimal order for each case is represented as a highlighted boxes. The optimal ARIMA(p,d,q) models for the sub-series $S_{1,n}^{COP-USD}, T_{1,n}^{COP-USD}, S_{1,n}^{Ecopetrol}$ *and* $T_{1,n}^{Ecopetrol}$ are, respectively: ARIMA(0,0,0), ARIMA(0,0,0), ARIMA(0,1,1) and ARIMA(2,0,2).

With the optimal ARIMA(p,d,q) models defined, the prediction can be performed for each sub-series, $S_{1,n}^{COP-USD}, T_{1,n}^{COP-USD}, S_{1,n}^{Ecopetrol}\ and\ T_{1,n}^{Ecopetrol}$, to obtain their respective prediction, that is: $S_{1,n+1}^{COP-USD}, T_{1,n+1}^{COP-USD}, S_{1,n+1}^{Ecopetrol}\ and\ T_{1,n+1}^{Ecopetrol}$. The lower part of figures 16 and 17 shows, respectively, the one-step prediction for sub-series of both USD-COP's exchange rate and Ecopetrol's stock.

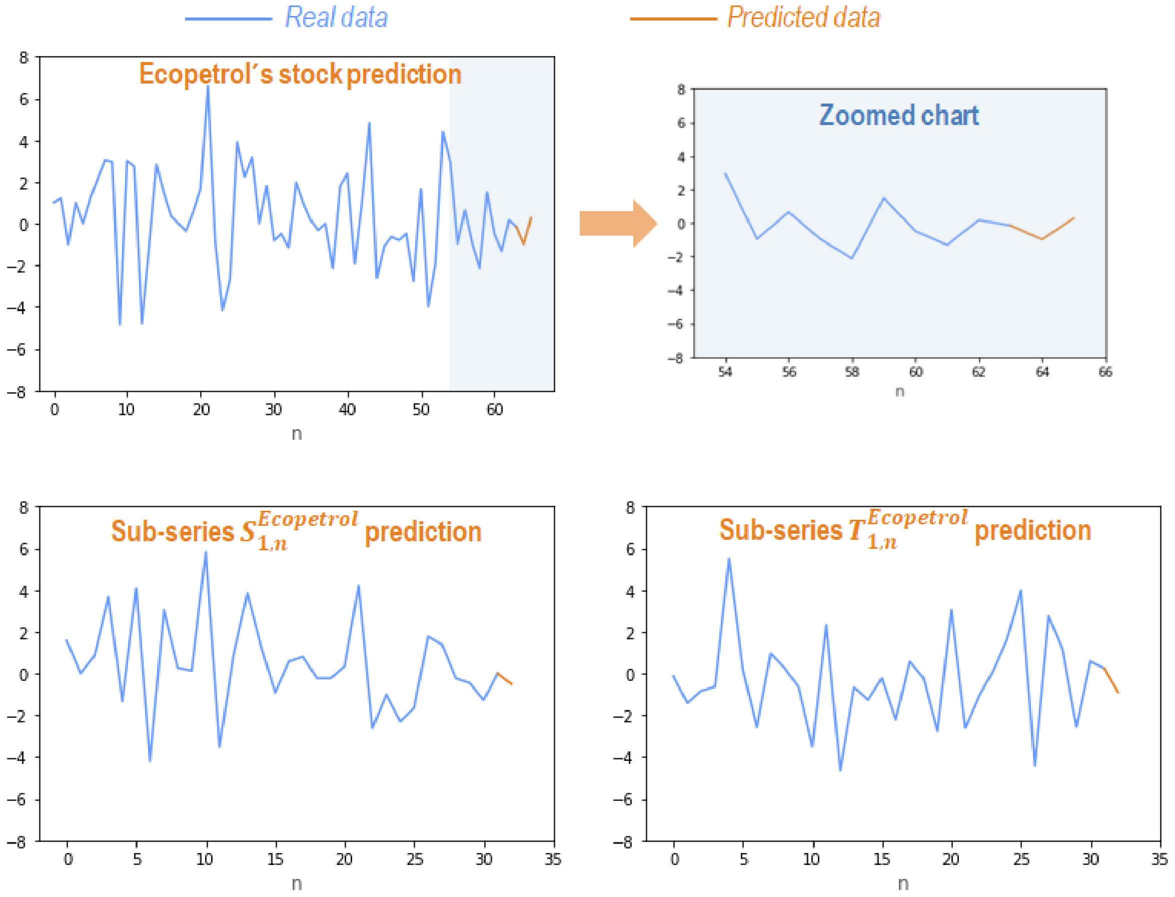
- **Third step: forecast of original series through the inverse discrete wavelet transform.** Once performed the prediction for the sub-series of both USD-COP's exchange rate and Ecopetrol's stock, the inverse discrete wavelet transform should be applied to obtain the two-step prediction for the original data. Since Daubechies D2 (Haar) was the mother wavelet used in the wavelet decomposition, this wavelet was also used to reconstruct the original series and hence to obtain the prediction. The upper part of figures 16 and 17 shows the two-step prediction for both log-returns series: USD-COP's exchange rate and Ecopetrol's stock.

Figure 16 Wavelet-based prediction of USD-COP's exchange rate log-returns



From the wavelet-based prediction of USD-COP's exchange rate log-returns, a near-zero return for the next two days (i.e. June 15 and 16, 2018) is expected, which can be better appreciated in the zoomed chart (upper right side).

Figure 17 Wavelet-based prediction of Ecopetrol's stock log-returns



From the wavelet-based prediction of Ecopetrol's stock log-returns, a negative return for the next day (i.e. June 15, 2018) is expected, but a positive one two days ahead (i.e. June 16, 2018), which can be better appreciated in the zoomed chart (upper right side).

IV.II Test of the wavelet-based prediction approach

The wavelet-based approach for prediction was performed as in previous section, and using moving 64-length windows of real data, to obtain 64 one-step predictions for both USD-COP's exchange rate and Ecopetrol's stock log-returns series. Likewise, a traditional ARIMA model was used to obtain 64 one-step predictions for the same both series. These one-step predictions are shown in figures 18 and 19 together with its respective squared error (SE) and mean squared error (MSE).

The results obtained suggest that traditional ARIMA model is slightly superior in prediction than the wavelet-based approach. The mean squared error of wavelet-based prediction for USD-COP's exchange rate log-returns is 0.8444 versus 0.6545 of traditional ARIMA model. Likewise, the mean squared error of wavelet-based prediction for Ecopetrol's stock log-returns is 5.1941 versus 4.6479 of traditional ARIMA model. Furthermore, looking at movement-direction prediction for USD-COP's exchange rate log-returns, the wavelet-based approach fails in 66% of days (42/64) versus 50% (32/64) of traditional ARIMA model. However, for Ecopetrol’s stock log-returns, the wavelet-based approach fails in 39% of days (25/64) versus 41% (26/64) of traditional ARIMA model.

Figure 18 One-step prediction of USD-COP’s exchange rate log-returns

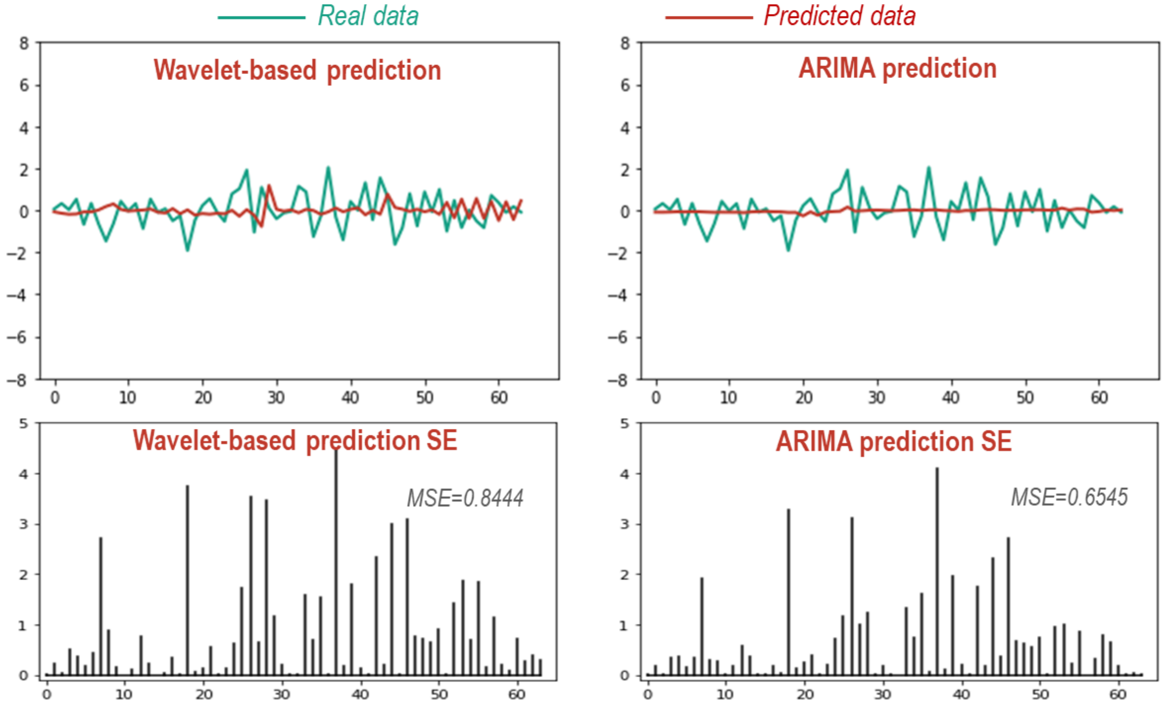
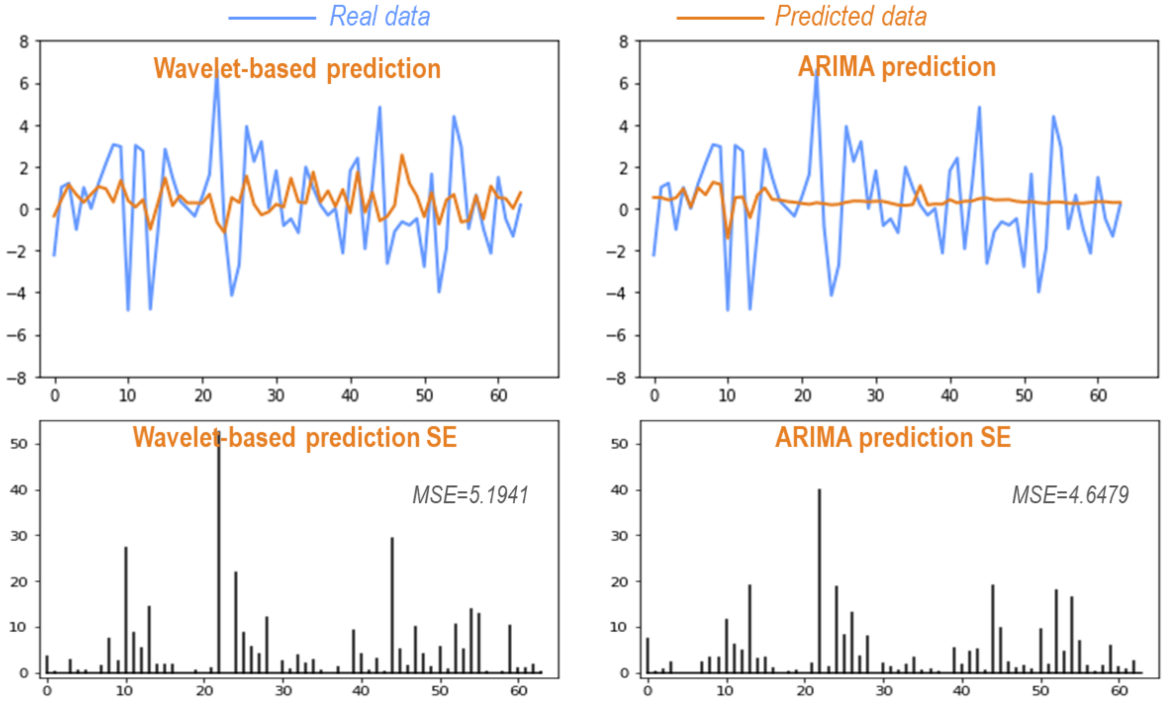


Figure 19 One-step prediction of Ecopetrol's stock log-returns



Concluding remarks

- The two basic ways in which a wavelet can be manipulated are by translation and scaling. Translation is referred to movements along the time axis and scaling is referred to the spreading out of the wavelet.

- The wavelet coefficients may be expressed as the inner product in the $L^2(\mathbb{R})$ space between the signal and the wavelet function, while the scaling coefficients may be expressed as the inner product between the signal and the wavelet function.

- The multiresolution representation refers to a representation of the signal expressed as the sum of the signal approximation and the signal detail at the next larger scale.

- The wavelet and scaling equations express, respectively, the wavelet and scaling function in terms of dilated and shifted versions of the associated scaling function.

- The wavelet and scaling equations lead to recursive equations for the approximation and wavelet coefficients which constitute the first half of the Fast Wavelet Transform also known as the multiresolution decomposition algorithm. In a similar way, the second half of the Fast Wavelet Transform is known as the reconstruction algorithm.

- The Haar is the simplest member of a family of discrete orthonormal wavelets known as the Daubechies wavelets. These wavelets satisfy the conditions required to build an orthogonal system and are very good representing polynomial behavior within a particular signal.

- The trade-off that should be considered when choosing the best wavelet for a particular data analysis is related to the number of scaling coefficients versus the support length of the wavelet.

- Daubechies scaling and wavelet functions have no explicit formulae except for the D2 (Haar). However, a discrete approximation of Daubechies scaling and wavelet functions can be generated through the reconstruction algorithm.

- Through the second half of the fast wavelet transform, the USD-COP's and Ecopetrol log-returns series can completely be recovered without losing information.

- The highest frequency oscillations are captured at the smallest scales and according to the signal details the largest fluctuations in USD-COP's exchange rate log-returns occur across the first three scales. Likewise, according to the signal details the largest fluctuations in Ecopetrol's stock log-returns occur across the first four scales towards the days 100 to 256 (i.e. from August 10, 2017 to June 14, 2018).

- For large oscillations of the analyzed signals, the D4 wavelet coefficients reach larger values than those reached using the Haar wavelet, which means that captures the high frequency oscillations of the signals better than the Haar wavelet.

- By using the D4 wavelet in the multiresolution decomposition, the signals are smoothed faster than by using the Haar wavelet because of the first one is relatively smoother.

- The wavelet-based approach presented in chapter IV predicts a near-zero return for the next two days (i.e. June 15 and 16, 2018) for USD-COP's exchange rate, and for Ecopetrol's stock, a negative return for the next day (i.e. June 15, 2018), but a positive for two days ahead (i.e. June 16, 2018).

- The traditional ARIMA model seems to be slightly superior in prediction than the wavelet-based approach. The mean squared error of wavelet-based prediction is larger than that one of traditional ARIMA model. Furthermore, looking at movement-direction prediction, the wavelet-based approach fails in more days than traditional ARIMA model for USD-COP's exchange rate but not for Ecopetrol's stock.

Ideas for future work

- Perform the wavelet-based approach proposed in chapter IV using another forecasting method instead of an ARIMA(p,d,q) model like a neural network, and compare its results with those obtained in here.

- Perform the wavelet-based approach proposed in chapter IV using another mother wavelet instead of a Daubechies D2 (Haar) like, for instance, a Daubechies D20, a Symmlet or a Coiflet and compare its results with those obtained in here.

- Explore others wavelet-based forecasting methods such as wavelet networks and compare its results with those obtained in here.

- Explore others wavelet-based methods to analyze financial time series such as static and dynamic correlations, causality relationships and contagion.

References

[1] GENCAY, R., SELCUK, F. AND WHITCHER, B. *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*. Orlando, Florida: Academic Press, 2002.

[2] ADISON, P. *The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance (2nd ed)*. Boca Raton, Florida: CRC Press, 2017.

[3] RENAUD, O., STARCK J-L. AND MURTAGH, F. *Prediction based on a multiscale decomposition*. International Journal of Wavelets, Multiresolution and Information Processing, 2003.

[4] NGUYEN, T. AND HE, T.-X. *Wavelet Analysis and Applications in Economics and Finance*. Reserch & Reviews: Journal of Statistics and Mathematical Sciences, 2015.

[5] GALLEGATI, M. AND SEMMLER, W. *Wavelet Applications in Economics and Finance: Dynamic Modeling and Econometrics in Economics and Finance*. Switzerland: Springer International Publishing, 2014.

[6] RANTA, M. *Wavelet Multiresolution Analysis of Financial Time Series*. Finland: Vaasan yliopisto, 2010.

[7] ALEXANDRIDIS, A., AND ACHILLEAS D. *Wavelet Neural Networks, With Applications in Financial Engineering, Chaos, and Classification*. New Jersey: John Wiley & Sons Inc, 2014.

[8] HIRSA, A. *Computational Methods in Finance*. London: CRC Press, 2013.

[9] HILPISCH, Y. *Python for Finance: Analyze big financial data*. United States of America: O'Reilly Media Inc, 2014.

[10] Colombian Stock Exchange. (2018). Obtained from https://www.bvc.com.co

[11] Quantcademy. (2018). QuantStart. Obtained from Basics of Statistical Mean Reversion Testing: https://www.quantstart.com

[12] Python. (2018). Python Software Foundation. Obtained from Seaborn: statistical data visualization: https://pypi.python.org

Appendix: Python scripts

#Wavelet function:

#$\Psi_{m,n}(\mathrm{t}) = \frac{1}{\sqrt{a_0^m}}\Psi\left(\frac{t-nb_o a_0^m}{a_0^m}\right)$ (1.6)

```python
def Psi_mn(Psi,t,a0,m,b0,n):
    return Psi(t*1.0/a0**m-n*b0)*a0**(-m*1.0/2)
```

#Wavelet coefficients for a scale-location grid of index m,n:

#$T_{m,n} = \sum_{t=0}^{N-1} x(t)\frac{1}{\sqrt{2^m}}\Psi\left(\frac{t-n2^m}{2^m}\right) = \langle x, \Psi_{m,n}\rangle$ (1.8)

```python
def T_mn(x,Psi,a0,m,b0,n):
    Psimn=[]
    T=int(len(x))
    for t in range(T):
        Psimn.append(Psi_mn(Psi,t,a0,m,b0,n))
    Tmn=dot(x,Psimn)
    return Tmn
```

#Reconstruction formula (inverse discrete wavelet transform):

#$x(t) = \sum_{m=1}^{M}\sum_{n=0}^{2^M-1} T_{m,n}\,\Psi_{m,n}\,(t)$ (1.9)

```python
def x_recons(x,Psi,t,a0,b0,A,B):
    xt=0.0
    N=int(len(x))
    M=int(log(N)*1.0/log(a0))
    for m in range(1,M+1):
        for n in range(N):
            xt+=T_mn(x,Psi,a0,m,b0,n)*Psi_mn(Psi,t,a0,m,b0,n)
    return xt*2.0/(A+B)
```

#Scaling function

#$\phi_{m,n}(t) = \frac{1}{\sqrt{2^m}}\phi\left(\frac{t-n2^m}{2^m}\right)$ (1.10)

```python
def Phi_mn(Phi,t,m,n):
    return Phi(t*1.0/2**m-n)*2**(-m*1.0/2)
```

#Approximation coefficients for a scale-location grid of index m,n

#$S_{m,n} = \sum_{t=0}^{2^M-1} x(t)\frac{1}{\sqrt{2^m}}\phi\left(\frac{t-n2^m}{2^m}\right) = \langle x, \phi_{m,n}\rangle$ (1.12)

```python
def S_mn(x,Phi,m,n):
    Phimn=[]
    T=int(len(x))
    for t in range(T):
        Phimn.append(Phi_mn(Phi,t,m,n))
    Smn=dot(x,Phimn)
    return Smn
```

#Continuous approximation (smooth version) of the signal at scale index m

#$x_m(t) = \sum_{n=0}^{2^{M-m}-1} S_{m,n}\, \phi_{m,n}(t)$ (1.13)

```python
def x_m(x,Phi,t,m):
    xm=0.0
    N=int(len(x))
    M=int(log(N)*1.0/log(2))
    for n in range(2**M-m):
        xm+=S_mn(x,Phi,m,n)*Phi_mn(Phi,t,m,n)
    return xm
```

#Signal detail at scale index m

#$d_m(t) = \sum_{n=0}^{2^{M-m}-1} T_{m,n}\, \Psi_{m,n}(t)$ (1.14)

```python
def d_m(x,Psi,t,a0,m,b0):
    dm=0.0
    N=int(len(x))
    M=int(log(N)*1.0/log(2))
    for n in range(2**M-m):
        dm+=T_mn(x,Psi,a0,m,b0,n)*Psi_mn(Psi,t,a0,m,b0,n)
    return dm
```

#Scaling or dilation equation

#$\phi(t) = \sum_{k=0}^{N_k-1} c_k \phi(2t-k)$ (1.16)

```python
def Phi_eq(Phi,c,t):
    Phit=0.0
    K=int(len(c))
    for k in range(K):
        Phit+=c[k]*Phi(2.0*t-k)
    return Phit
```

#Wavelet equation

#$\Psi(t) = \sum_{k=0}^{N_k-1} b_k \phi(2t-k)$ (1.20)

```python
def Psi_eq(Phi,c,t):
    Psit=0.0
    Nk=int(len(c))
    for k in range(Nk):
        b=((-1)**k)*c[Nk-1-k]
        Psit+=b*Phi(2.0*t-k)
    return Psit
```

#Scaling function in terms of shifted and dilated versions of itself

#$\phi_{m+1,n}(t) = \frac{1}{\sqrt{2}}\sum_{k=0}^{N_k-1} c_k \phi_{m,2n+k}(t)$ (1.21)

```python
def Phi_recursive(Phi,c,m,n):
    Phimnextn=0.0
    K=int(len(c))
    for k in range(K):
        Phimnextn+=c[k]*Phi(t*1.0/2**(m-1)-2*n-k)
    return Phimnextn*2**(-0.5)
```

#Wavelet function in terms of shifted and dilated versions of its scaling function

#$\Psi_{m+1,n}(t) = \frac{1}{\sqrt{2}}\sum_{k=0}^{N_k-1} b_k \phi_{m,2n+k}(t)$ (1.22)

```python
def Psi_recursive(Phi,c,m,n):
    Psimnextn=0.0
    Nk=int(len(c))
    for k in range(Nk):
        b=((-1)**k)*c[Nk-1-k]
        Psimnextn+=b*Phi(t*1.0/2**(m-1)-2*n-k)
    return Psimnextn*2**(-0.5)
```

#First half of the FWT: multiresolution decomposition algorithm

##Recursive method for approximations coefficients for a scale-location grid of index m,n

##$S_{m+1,n} = \frac{1}{\sqrt{2}}\sum_{k=0}^{N_k-1} c_k S_{m,2n+k}$ (1.23)

```python
def Smnext_n(x,Phi,c,m,n):
    Smnextn=0.0
    K=int(len(c))
    for k in range(K):
        Smnextn+=c[k]*S_mn(x,Phi,m-1,2.0*n+k)#Lowpass filter: c[k]*2**(-0.5)
    return Smnextn*2**(-0.5)
```

##Recursive method for wavelet coefficients for a scale-location grid of index m,n

##$T_{m+1,n} = \frac{1}{\sqrt{2}}\sum_{k=0}^{N_k-1} b_k S_{m,2n+k}$ (1.24)

```python
def Tmnext_n(x,Phi,c,m,n):
    Tmnextn=0.0
    Nk=int(len(c))
    for k in range(Nk):
        b=((-1)**k)*c[Nk-1-k]
        Tmnextn+=b*S_mn(x,Phi,m-1,2.0*n+k)#Highpass filter: b[k]*2**(-0.5)
    return Tmnextn*2**(-0.5)
```

#Second half of the FWT: reconstruction algorithm

$$\#S_{m-1,n} = \frac{1}{\sqrt{2}}\left[\sum_{k=0}^{2^{M-m-1}-1} c_{n-2k} S_{m,k} + \sum_{k=0}^{2^{M-m-1}-1} b_{n-2k} T_{m,k}\right] \qquad (1.25)$$

```python
def Smprev_n(x,c,n):
    Smprevn=0.0
    K=int(len(x)*1.0/2)
    Nk=int(len(c))
    c2,b=0,0
    for k in range(K):
        if((n-2*k)>=0 and (n-2*k)<Nk):
            c2=c[n-2*k]
            b=((-1)**(n-2*k))*c[Nk-1-(n-2*k)]
        else:
            c2,b=0,0
        Smprevn+=c2*x[k]+b*x[k+K]
    return Smprevn*2**(-0.5)
```

#Evaluating an ARIMA(p,d,q) model

```python
def evaluate_arima(X,pdq_order):
    #Training dataset
    train_n=int(len(X)*2.0/3.0)
    train,test=X[0:train_n],X[train_n:]
    history=[x for x in train]
    #Predictions
    predictions=list()
    for t in range(len(test)):
        model=ARIMA(history,order=pdq_order)
        model_fit=model.fit(disp=0)
        x_hat=model_fit.forecast()[0]
        predictions.append(x_hat)
        history.append(test[t])
    #Out of sample error
    error=mean_squared_error(test,predictions)
    return error
```